# Ramping Up Open-Source RISC-V Cores: Assessing the Energy Efficiency of Superscalar, Out-of-Order Execution


Integrated Systems Laboratory (ETH Zürich)

**Zexin Fu**[1] zexifu@iis.ee.ethz.ch
**Riccardo Tedeschi**[2] riccardo.tedeschi6@unibo.it
**Gianmarco Ottavi**[2] gianmarco.ottavi2@unibo.it
**Nils Wistoff**[1] nwistoff@iis.ee.ethz.ch
**César Fuguet**[3] cesar.fuguet@univ-grenoble-alpes.fr
**Davide Rossi**[2] davide.rossi@unibo.it
**Luca Benini**[12] lbenini@iis.ee.ethz.ch

[1] IIS, ETH Zurich, Switzerland
[2] DEI, University of Bologna, Italy
[3] Univ. Grenoble Alpes, Inria, TIMA, France


**PULP Platform**
Open Source Hardware, the way it should be!

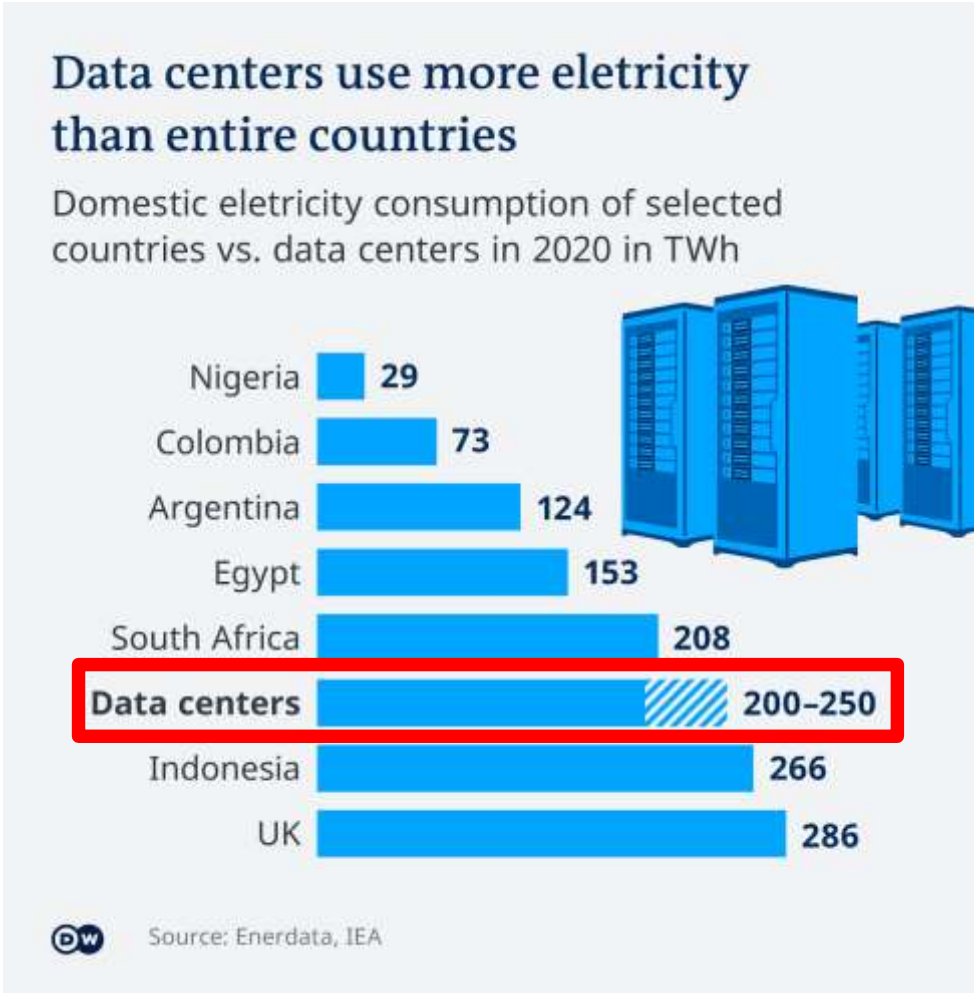
@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

# Energy Efficiency: A Rising Concern

**Energy Efficiency ($\frac{\text{Operations}}{\text{Power}\cdot\text{Time}}$) Matters Everywhere:**

- Low-power Embedded/IoT systems:
  - Battery life
  - Thermal limit (Passive cooling)
- High-performance computing (HPC):
  - Thermal limit (Thermal Design Power)
  - Total cost of ownership

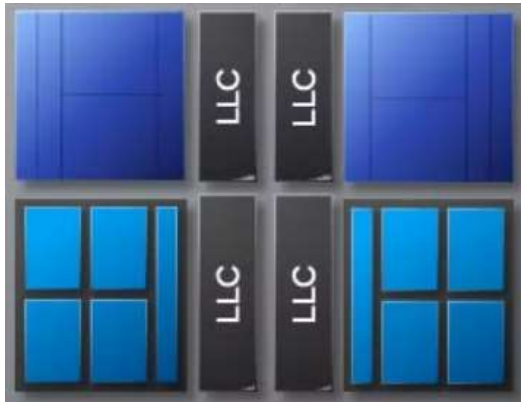

Nano-size UAV drones



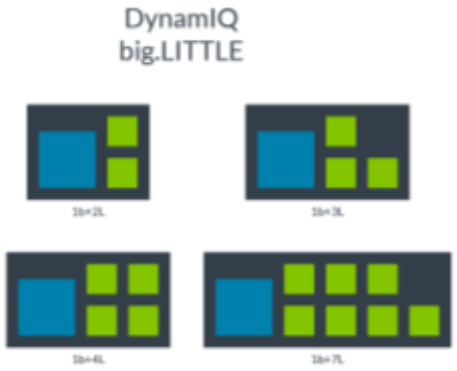

Data center electricity consumption

# Energy Efficiency: A Rising Concern

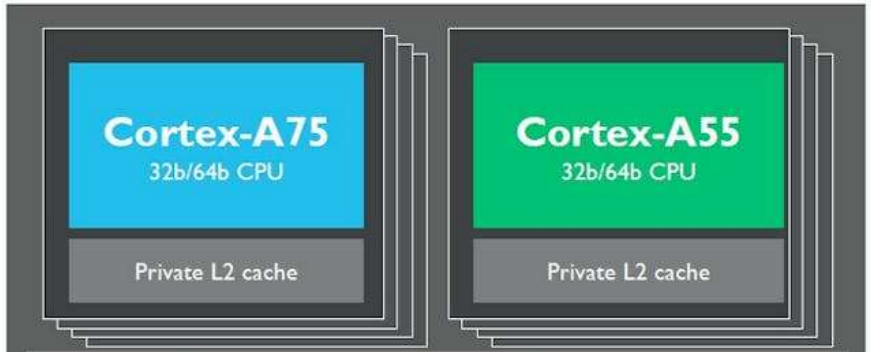
- **Prevailing Energy Efficient Design: Heterogeneous CPU cluster**
  - High Performance Core + Low Power Core
  - Balance **Peak Performance** with Power-Efficient Handling of **Lightweight Tasks**




Intel Performance Cores and Efficient Cores




ARM big.LITTLE design




| ARM Cortex-A75 | ARM Cortex-A55 |
|---|---|
| Decode width: 3 | Decode width: 2 |
| Out-of-order Issue | In-order Issue |
| Pipeline stages: 11-13 | Pipeline stages: 8 |

ARM big.LITTLE
Example: A75 + A55

# Energy Efficiency: A Rising Concern

## Big Core

**ARM Cortex-A75**

Decode width: **3**

**Out-of-order** Issue

Pipeline stages: **11-13**

## Little Core

**ARM Cortex-A55**

Decode width: **2**

**In-order** Issue

Pipeline stages: **8**

# Energy Efficiency: A Rising Concern

## Big Core

**Core A**

Decode width: **Wide**

**Out-of-order** Issue

Pipeline stages: **>10**

## Little Core

**Core B**

Decode width: **Narrow**

**In-order** Issue

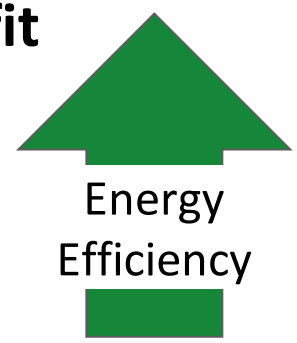
Pipeline stages: **3~10**

**Which core offers better Energy Efficiency ?**

# Superscalar & OoO Cores are Less Energy Efficient?

- **The belief of inherently inefficiency of Superscalar / OoO cores…**
  - Complex & power-hungry scheduling hardware
  - Speculative execution overheads
  - Typically deeper pipeline and higher frequency


- **But with superscalar / OoO (and other HW structs), we have the benefit**
  - Higher IPC & throughput, shorter execution time
  - Better branch prediction to reduce the miss penalty
  - Higher frequency, shorter execution time




- Is **Superscalar / OoO** always **Less Energy Efficient**?
- How much we can push the **Core Performance** before we become **Truly Inefficient** ?

# Contributions

*We conduct a detailed and **fair** comparative **analysis** of **Performance**, **Area**, **Power**, and **Energy Efficiency***

**Three open-source cores w/**

**different µ-arch**

- **CVA6**
- **CVA6S+**
- **C910**

**FAIR**

**Comparison**

- *Base **ISA** (RV64IMAFDC)*
- ***SoC** environment*
- ***Technology** node (GF22 FDX)*
- ***EDA** tools*
- *Implementation **methodology***
- ***Benchmark** binaries*

```
github.com/pulp-platform/cva6
github.com/pulp-platform/pulp-c910
```

# It’s NOT Easy to Assess & Compare Energy Efficiency

- **Why not commercial cores**
  - **Closed-source** black boxes
  - Different **ISAs and Extensions**
  - Different **Platforms and IPs**
  - Different **Technology and Methodologies**
  - Different **Benchmark Suites, Compilation Tools and Flags**

# It’s NOT Easy to Assess & Compare Energy Efficiency

## **Open-Source** RISC-V Cores to the Rescue!

- **CVA6**[1] **Single** Decode Width, **In-Order**, **6**-stage RV64 Core
- **CVA6S+** **Dual** Decode Width, **In-Order**, **6**-stage RV64 Core
- **C910**[2] **Three** Decode Width, **Out-of-Order**, **12**-stage RV64 Core

All Written in **Verilog/SystemVerilog**:
Backend Tools Friendly!

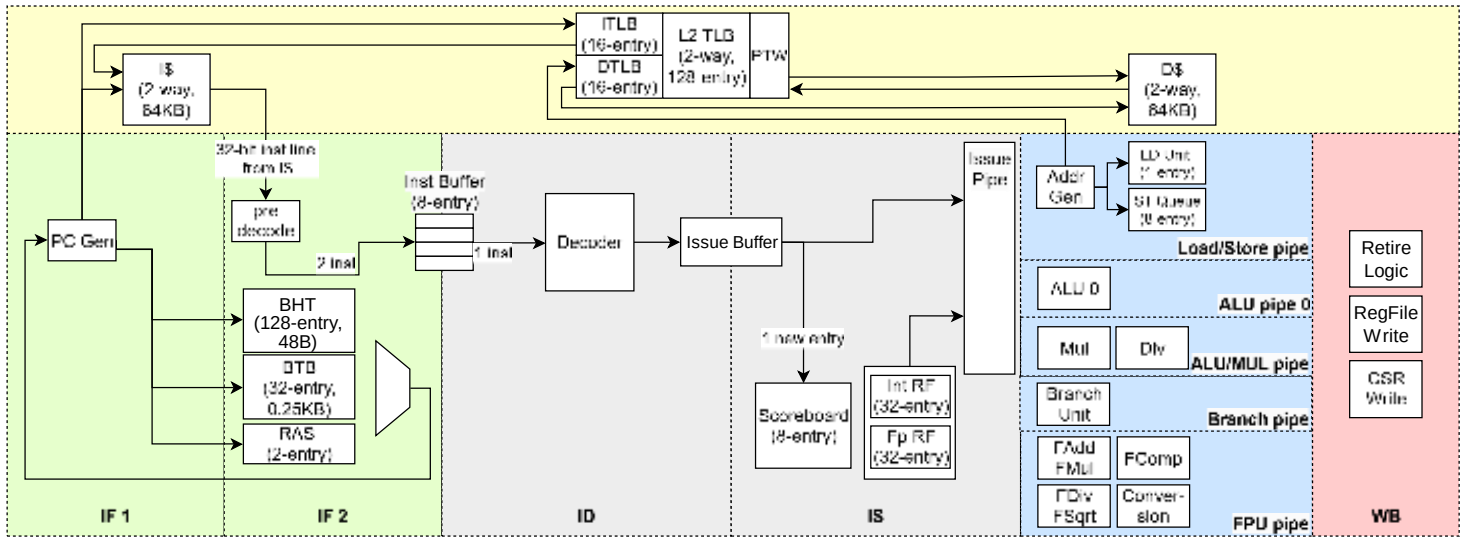

[1] F. Zaruba and L. Benini, "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology", IEEE VLSI, 2019
[2] Chen, C, et al. "Xuantie-910: A commercial multi-core 12-stage pipeline out-of-order 64-bit high performance RISC-V processor with vector extension: Industrial product ", ACM/IEEE ISCA, 2020

# Next, Let's go into the Details of their µ-architecture Difference

# Micro-architecture: **CVA6**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- RV64IMAFDC, Linux Capable;
- 6-stage, Single-issue, In-order;
- **IF**: 32-bit instruction fetch;
- **BP**: Basic 128-entry BHT, 32-entry BTB, and 2-entry RAS;
- **Scoreboard**: 8-entry, track in-flight instr, data dependency;
- **Commit**: Up to 2 instrs per cycle

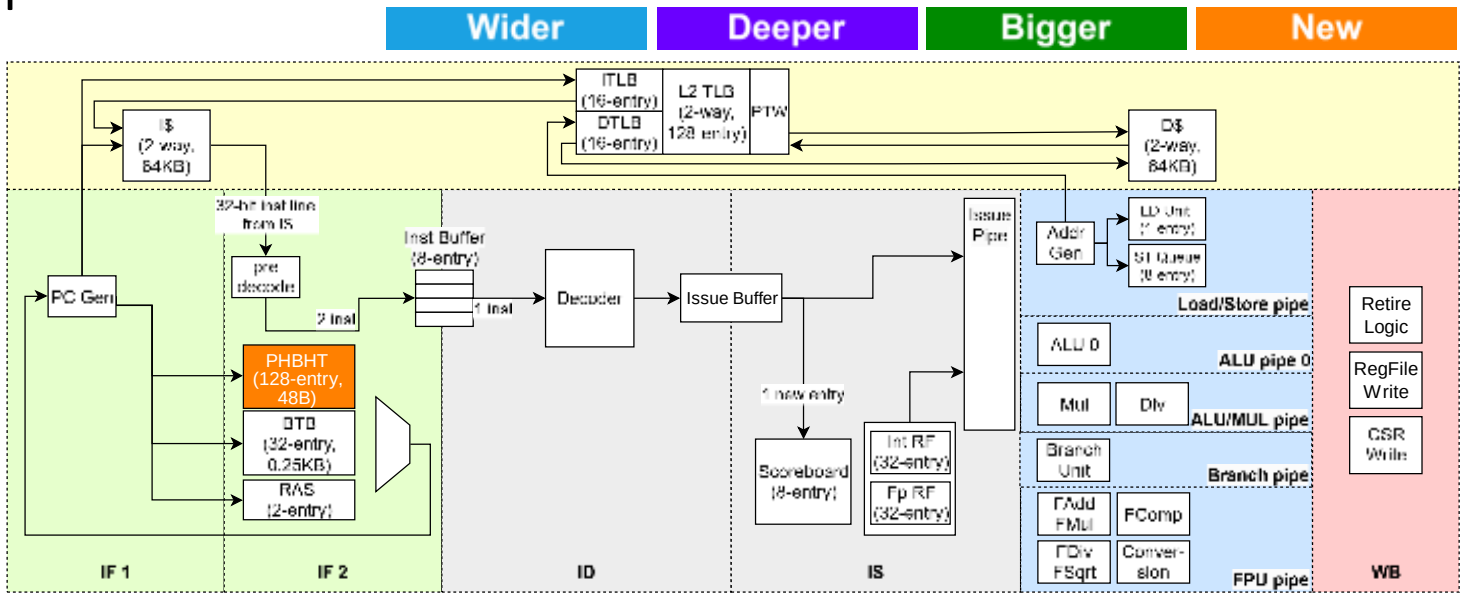



CVA6 µArchitecture

# Micro-architecture: CVA6 → CVA6S+

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Better BP**: Private History Branch History Table (PHBHT) predictor

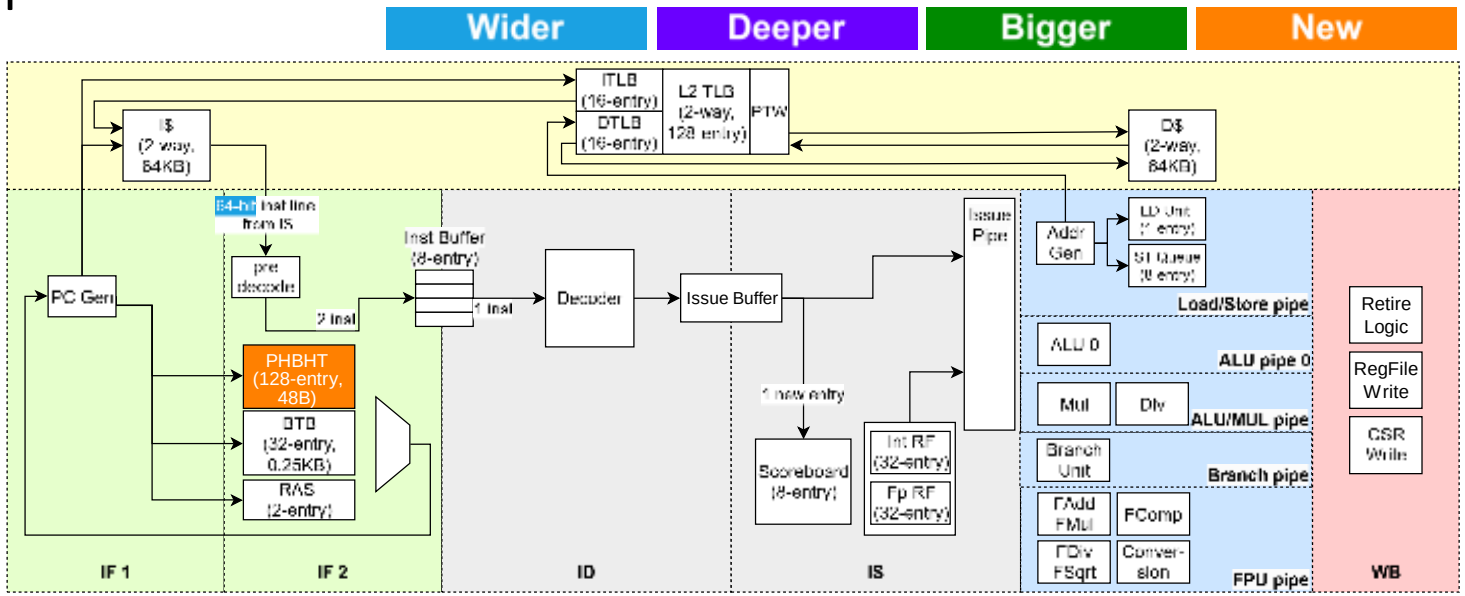



CVA6S+ µArchitecture

# Micro-architecture: CVA6 → CVA6S+

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Better BP**: Private History Branch History Table (PHBHT) predictor
- **Wider IF**: 64-bit fetch, up to 4 instructions;

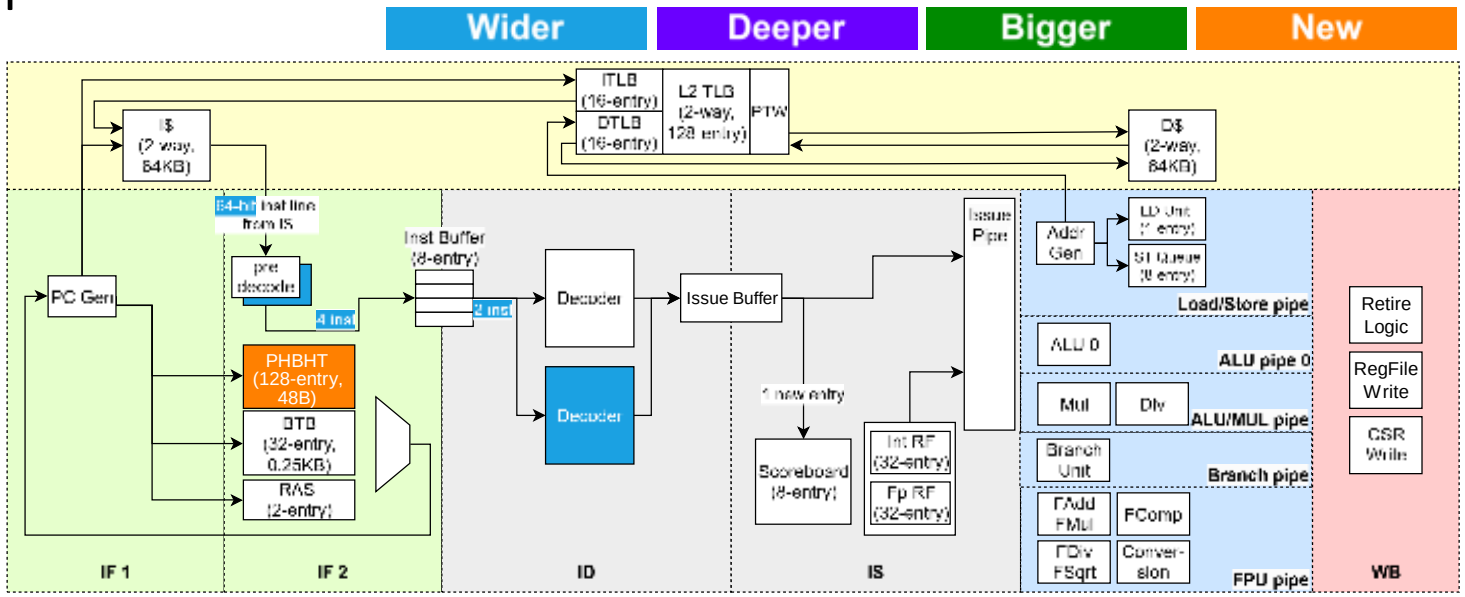



CVA6S+ µArchitecture

# Micro-architecture: CVA6 → CVA6S+

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Better BP**: Private History Branch History Table (PHBHT) predictor
- **Wider IF**: 64-bit fetch, up to 4 instructions;
- **Wider ID**: 2 decoders;

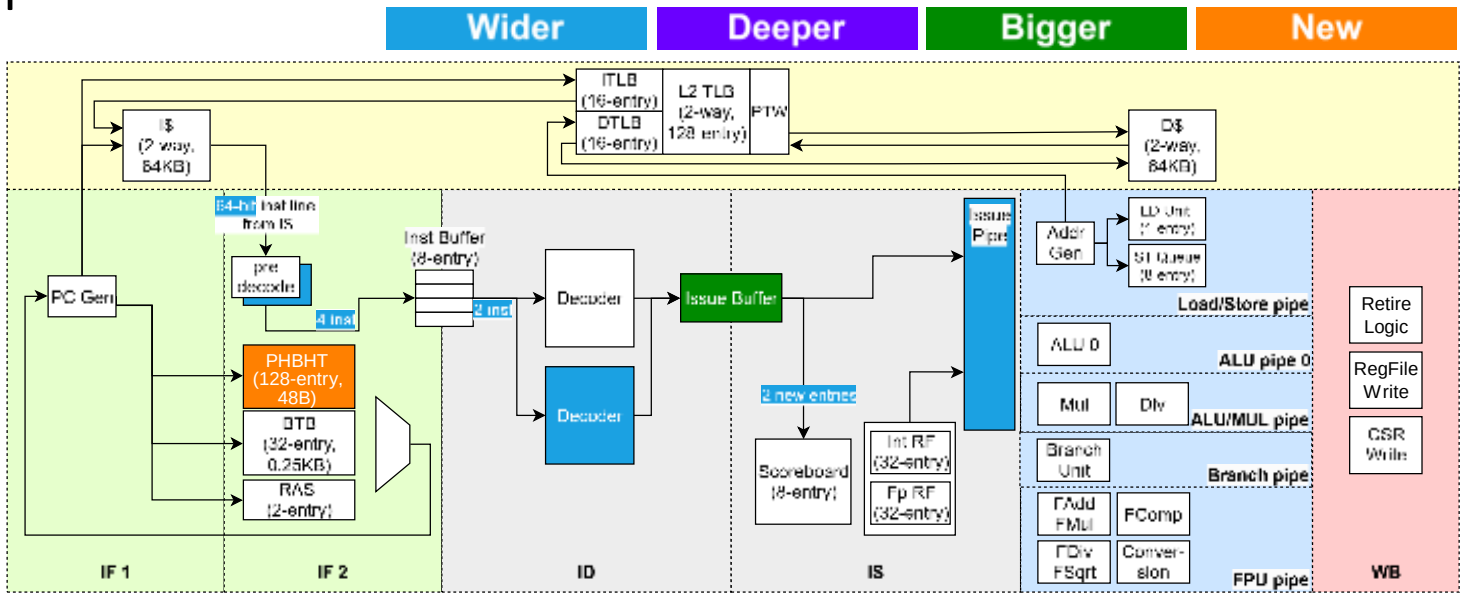



CVA6S+ µArchitecture

# Micro-architecture: CVA6 → CVA6S+

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Better BP**: Private History Branch History Table (PHBHT) predictor
- **Wider IF**: 64-bit fetch, up to 4 instructions;
- **Wider ID**: 2 decoders;
- **Wider IS**: 2 issue width;

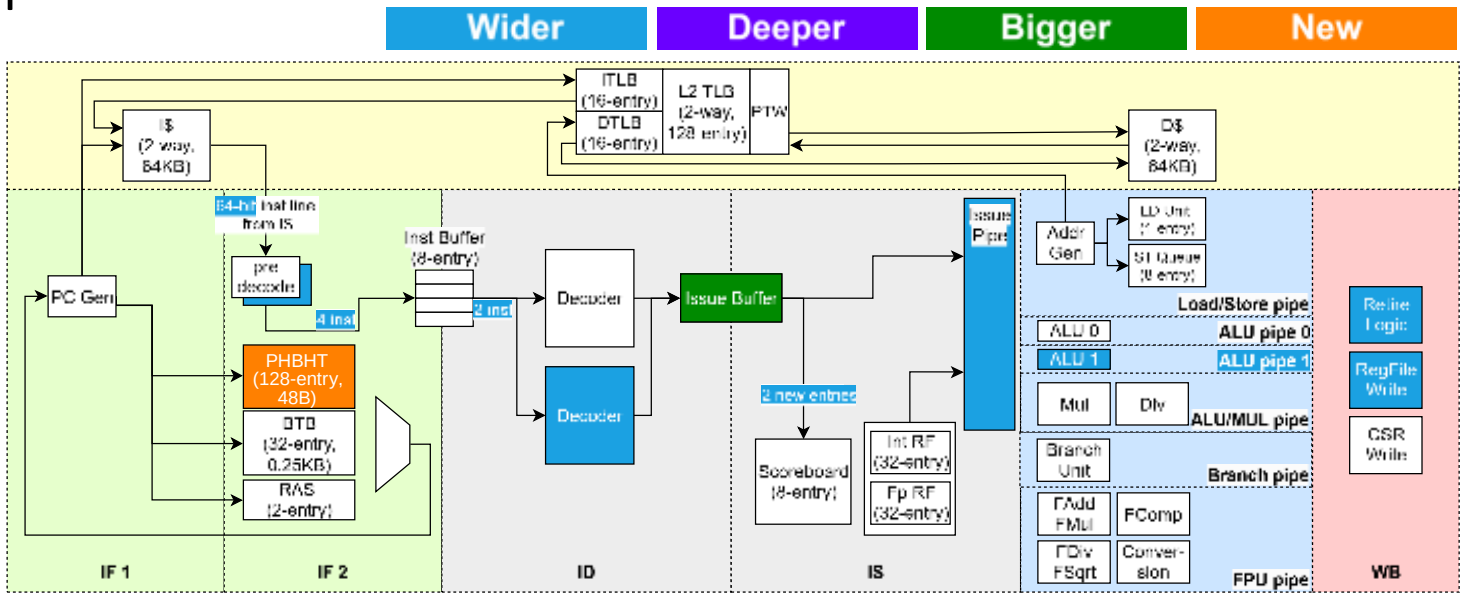



CVA6S+ µArchitecture

# Micro-architecture: CVA6 → CVA6S+

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

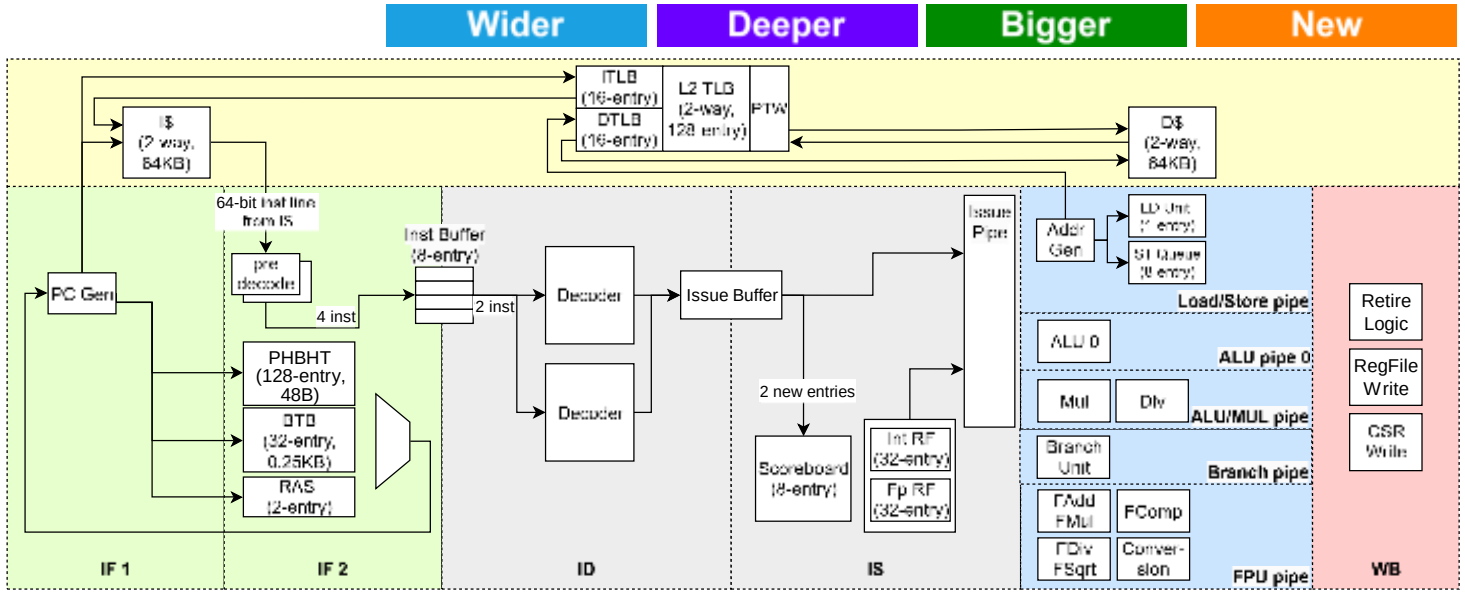
- **Better BP**: Private History Branch History Table (PHBHT) predictor
- **Wider IF**: 64-bit fetch, up to 4 instructions;
- **Wider ID**: 2 decoders;
- **Wider IS**: 2 issue width;
- **Wider EX**: 2 ALUs + ALU-to-ALU forwarding




CVA6S+ µArchitecture

# Micro-architecture: **C910**

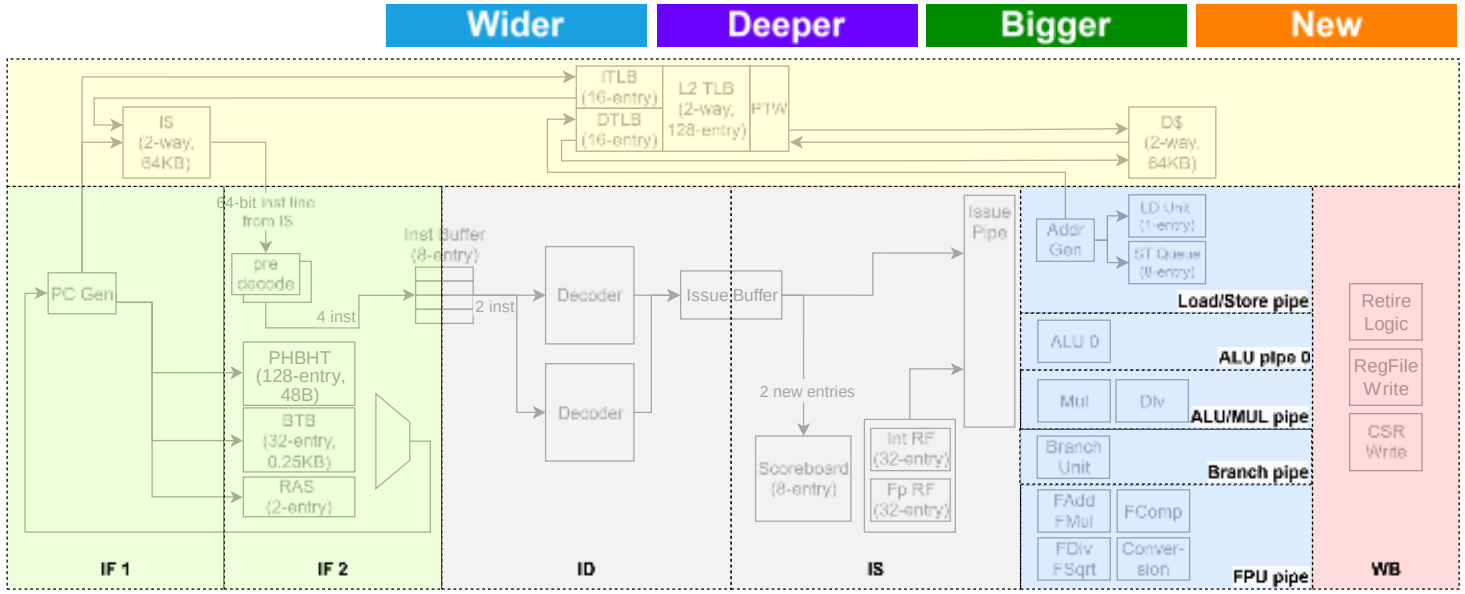
| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |




CVA6S+ µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

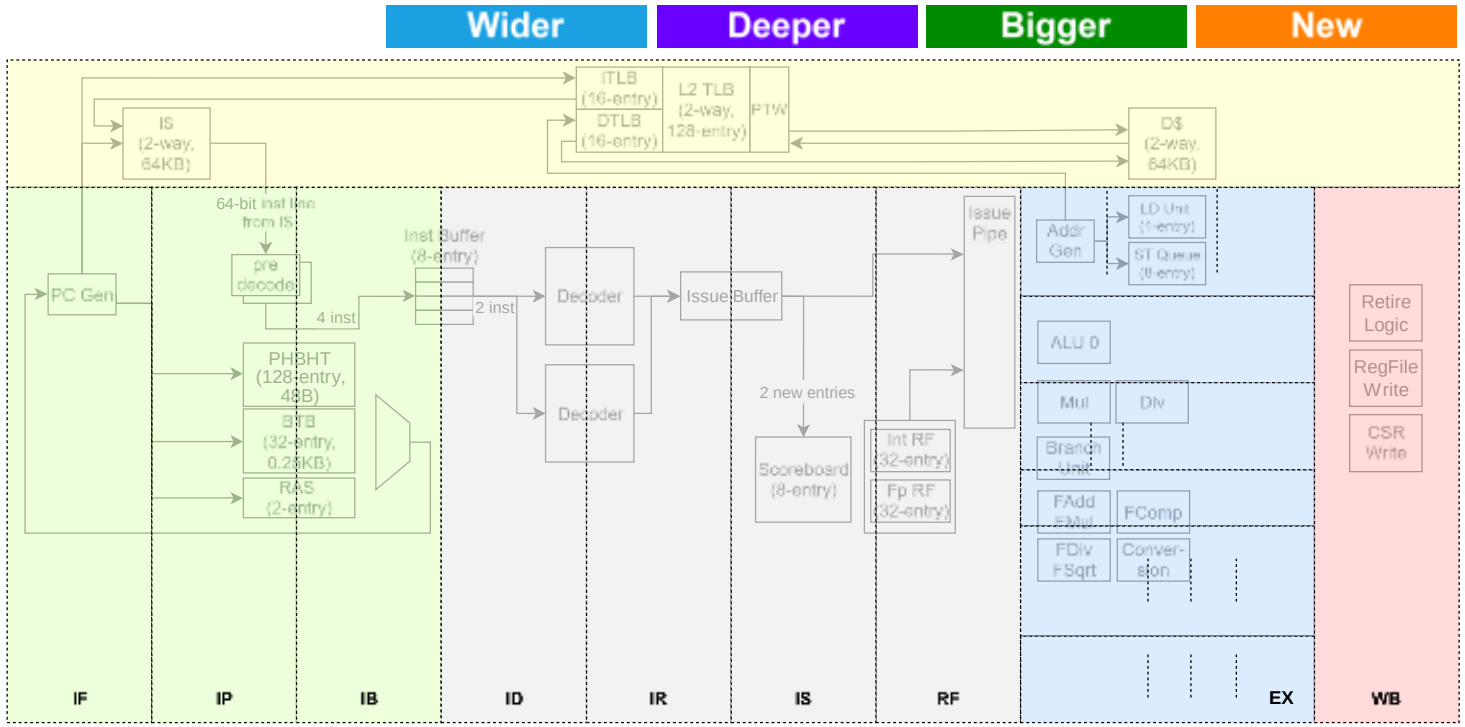



CVA6S+ µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Deeper Pipeline:** 12 stages

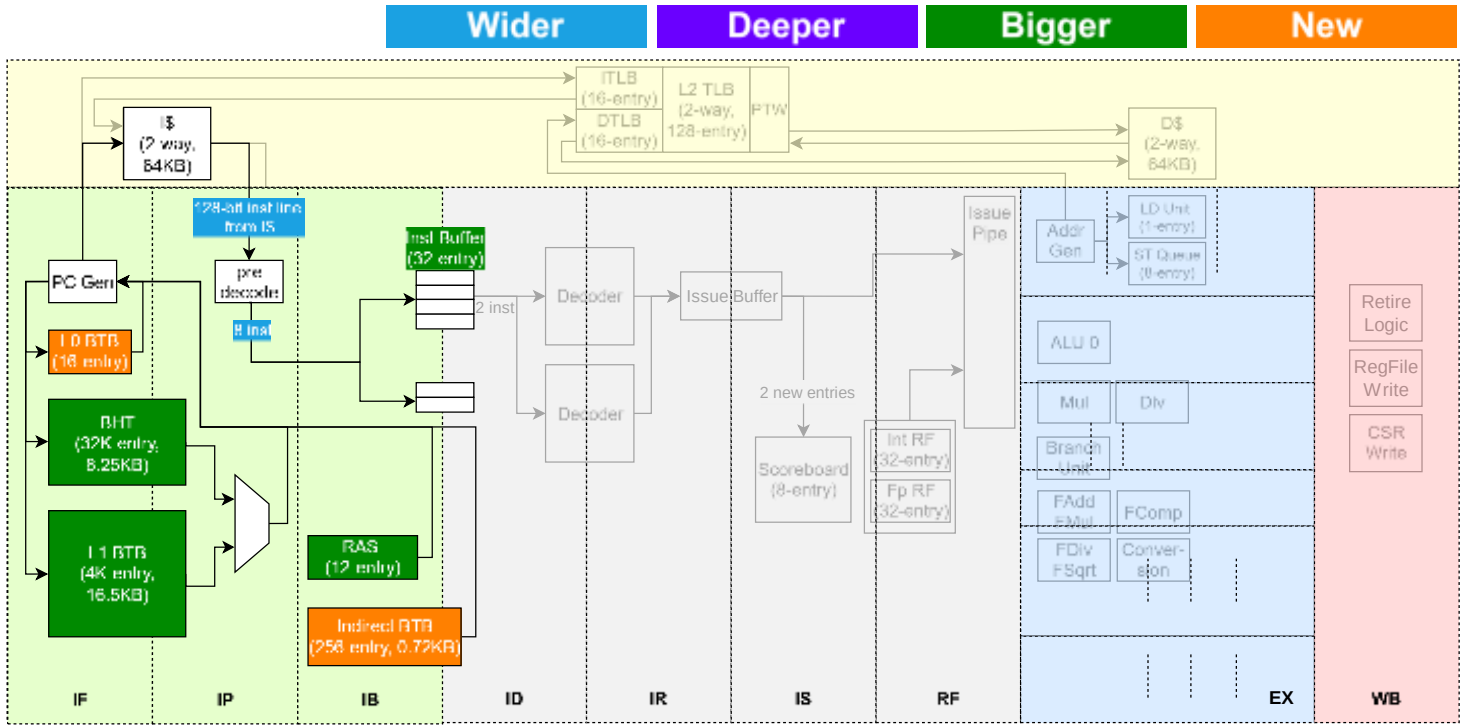



C910 µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Deeper Pipeline:** 12 stages
- **Wider IF**: 128-bit fetch
- **Advanced Branch Prediction:**
  - 32K-entry BHT
  - 16-entry L0 BTB + 4K-entry L1 BTB + 256-entry Indir BTB
  - 12-entry RAS

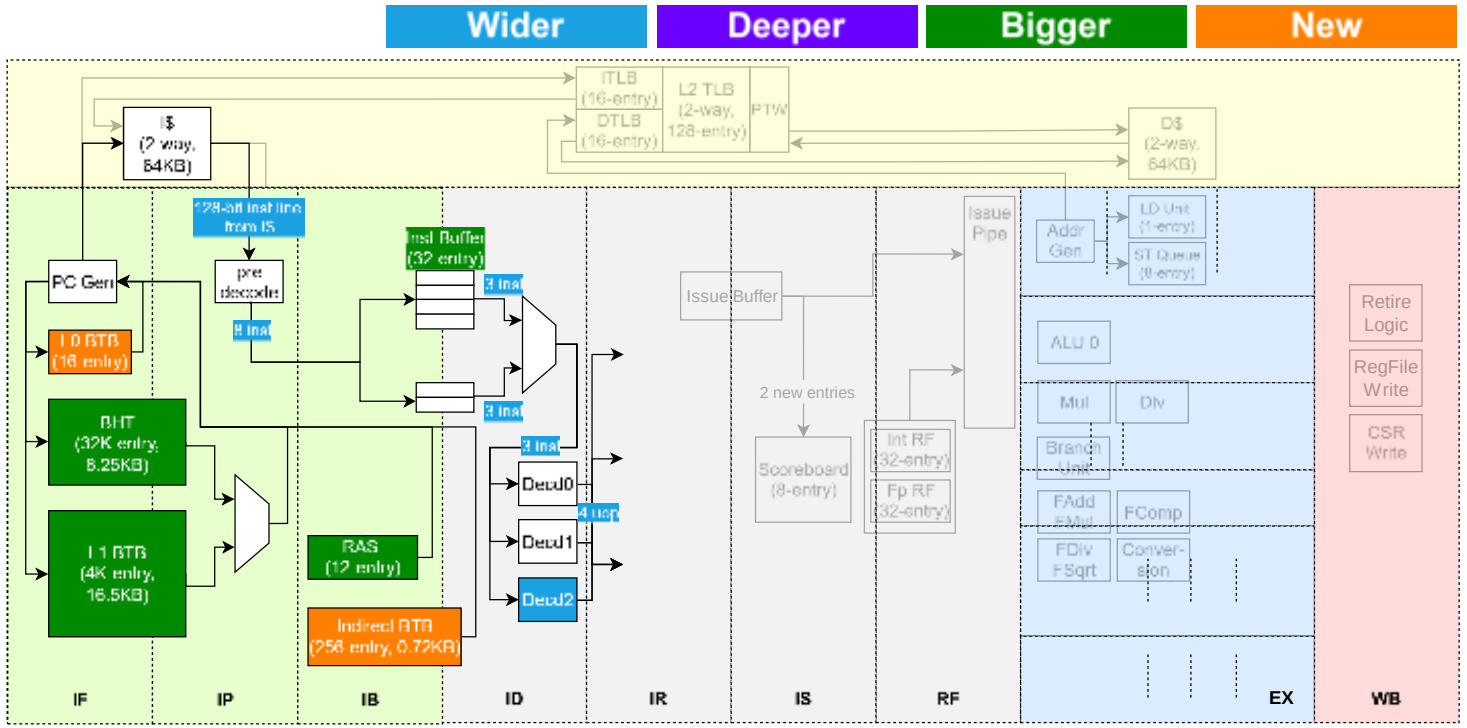



C910 µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Deeper Pipeline:** 12 stages
- **Wider IF**: 128-bit fetch
- **Advanced Branch Prediction:**
  - 32K-entry BHT
  - 16-entry L0 BTB + 4K-entry L1 BTB + 256-entry Indir BTB
  - 12-entry RAS
- **Wider ID**: 3 decoders

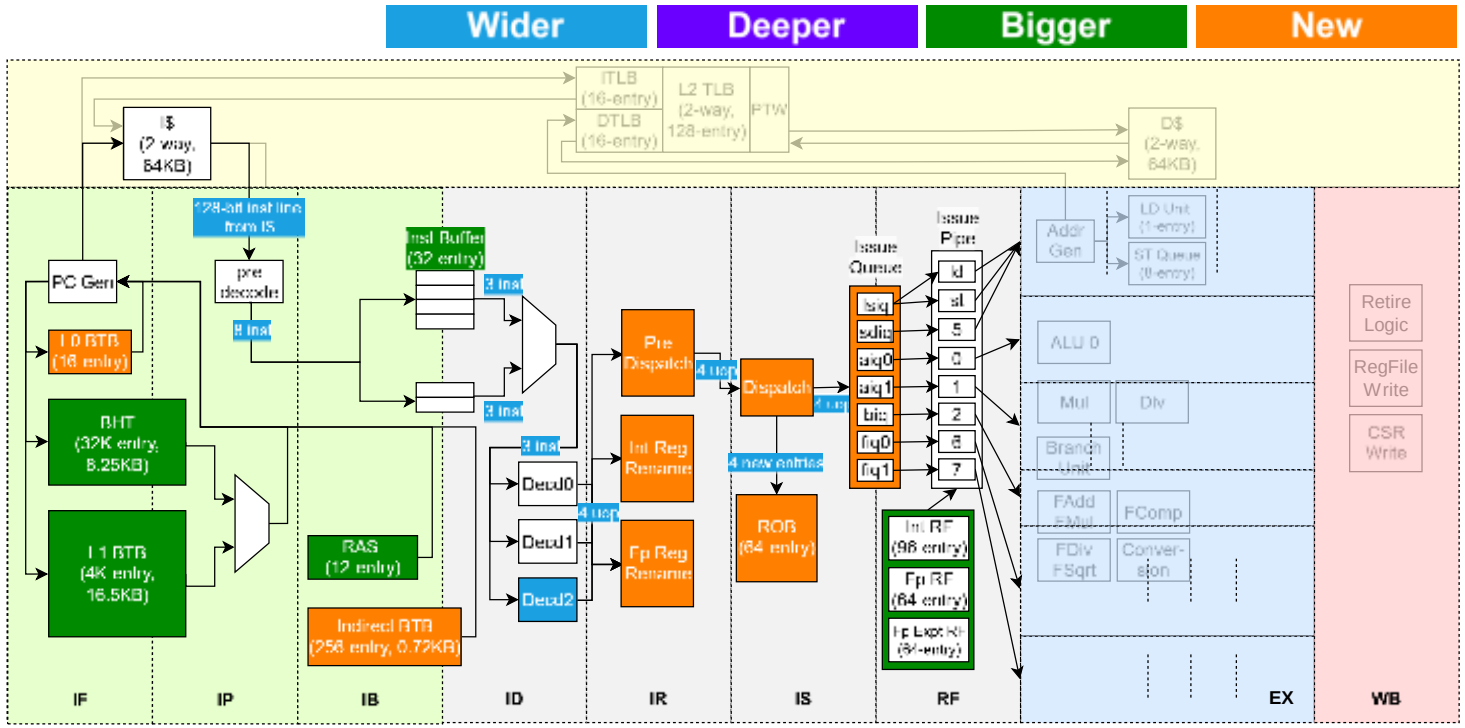



C910 µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Deeper Pipeline:** 12 stages
- **Wider IF**: 128-bit fetch
- **Advanced Branch Prediction:**
  - 32K-entry BHT
  - 16-entry L0 BTB + 4K-entry L1 BTB + 256-entry Indir BTB
  - 12-entry RAS
- **Wider ID**: 3 decoders
- **Out-of-Order Support**
- **Wider IS**: 8 issue pipe
- **Larger Buffers & Register Files:**
  - 64-entry ROB
  - 96-entry physical regfile

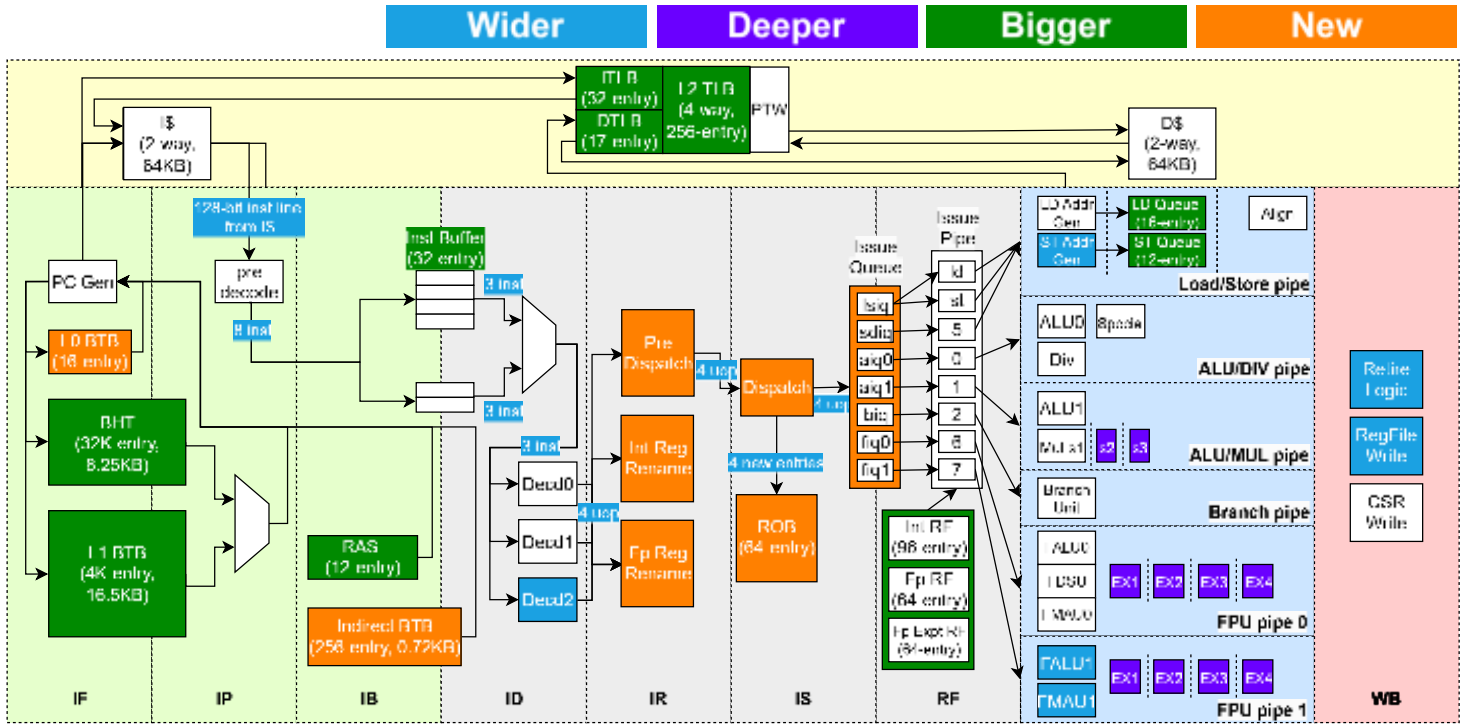



C910 µArchitecture

# Micro-architecture: **C910**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Deeper Pipeline:** 12 stages
- **Wider IF**: 128-bit fetch
- **Advanced Branch Prediction:**
  - 32K-entry BHT
  - 16-entry L0 BTB + 4K-entry L1 BTB + 256-entry Indir BTB
  - 12-entry RAS
- **Wider ID**: 3 decoders
- **Out-of-Order Support**
- **Wider IS**: 8 issue pipe
- **Larger Buffers & Register Files:**
  - 64-entry ROB
  - 96-entry physical regfile
- **Function Units:**
  - 2 Int pipes
  - 2 FP pipes
  - 1 Branch pipe
  - LSU: ld/st pipes




C910 µArchitecture

# Take away

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **The key µ-arch complexity difference of three cores**
  1. Branch predictor **CVA6 < CVA6S+ << C910**
  2. Scoreboard/ROB **CVA6 ≈ CVA6S+ << C910**
  3. Issue stage **CVA6 ≈ CVA6S+ << C910**
  4. Execution stage **CVA6 < CVA6S+ << C910**
  5. Load-store Unit **CVA6 = CVA6S+ << C910**

# Platform: Cheshire[1]

- **Modular Platform**
  - Linux-capable and configurable for application-class RISC-V cores.
  - Standard interrupt, debug, and memory interfaces
- **Integrate all three cores into Cheshire**
  - Same memory hierarchy and bandwidth
  - Same timer and interrupt supports;
  - Same peripherals;
  - Same software stack.

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

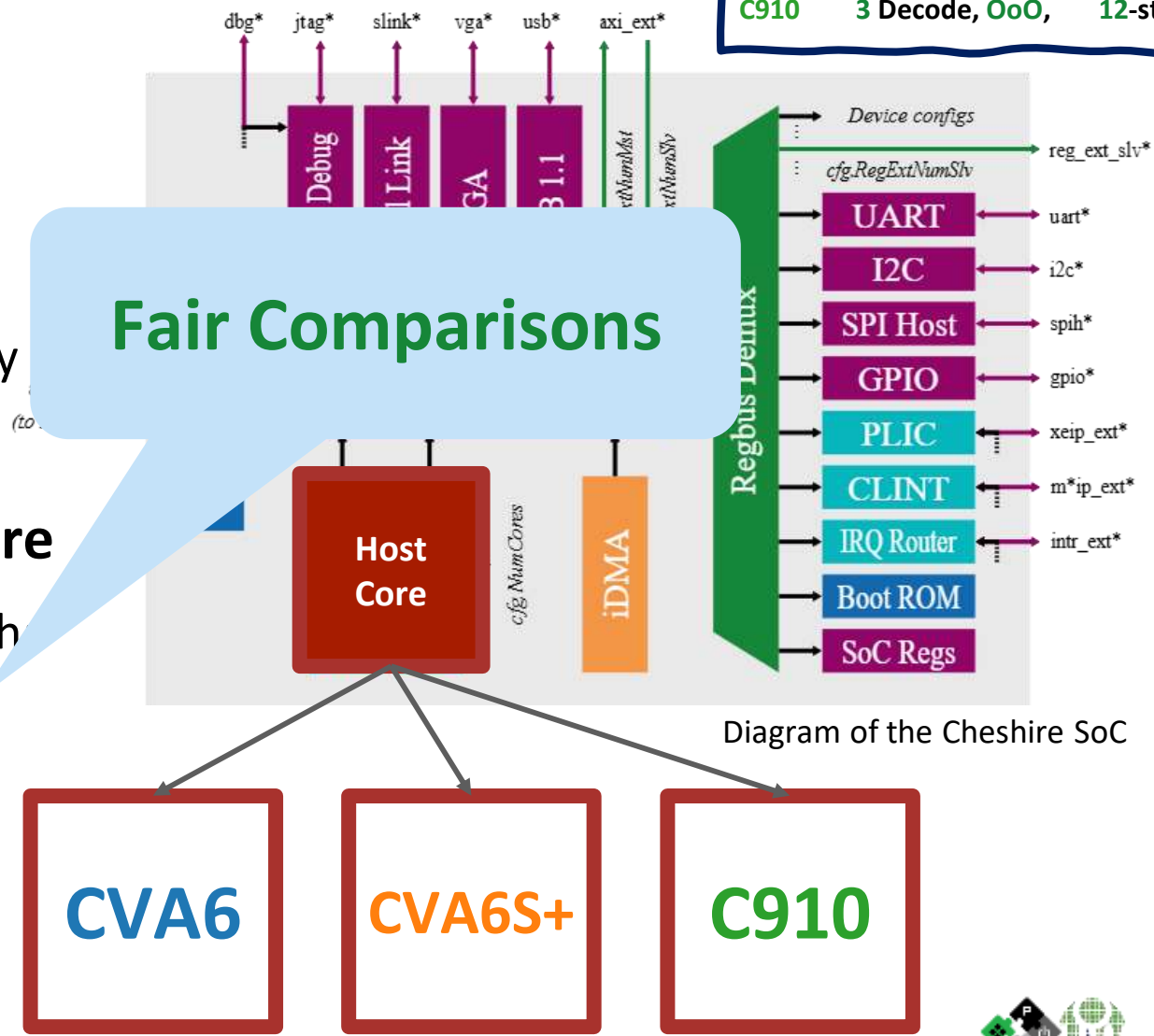
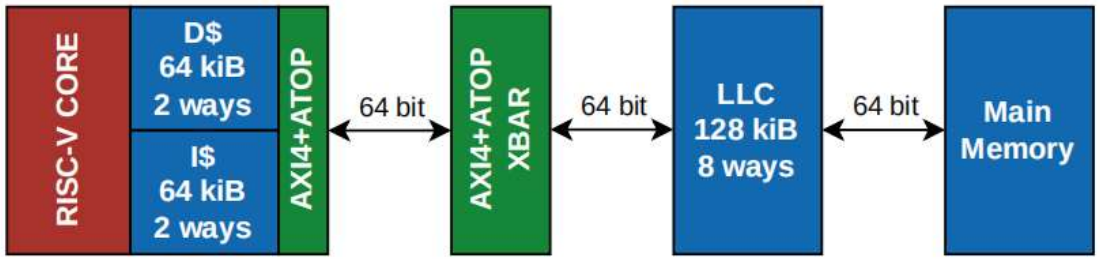



Diagram of the Cheshire SoC

[1] Ottaviano , et al. "Cheshire: A lightweight, linux-capable risc-v host platform for domain-specific accelerator plug-in". IEEE TCAS, 2023

# Memory Subsystem

- **Align memory subsystem as much as possible**
  - L1 Cache (Integration with LSUs per core)
    - Data cache: 64 KB, 2 ways
    - Instr cache: 64 KB, 2 ways
  - Last-level Cache (On Cheshire SoC)
    - 128 KB, 8 ways
  - Memory Interface
    - 64-bit width AXI




Memory hierarchy of the Cheshire SoC

# Next, Let's Take a Look at the Performance Benchmarking

# Software Benchmark Setup

- **Suits**
  - CoreMark
    - **Industry-standard** core performance metric.
    - List processing, matrix manipulation, state machine, CRC
  - Embench-IoT Suite
    - **Embedded/IoT** applications
  - RaiderSTREAM Suite
    - Adapted from the STREAM benchmark.
    - **Memory bandwidth** benchmark
    - Sequential & irregular access (cold cache)
- **Environment**
  - Bare metal: No OS overhead, used for both IPC and power
  - RTL simulation: QuestaSim 2021.3

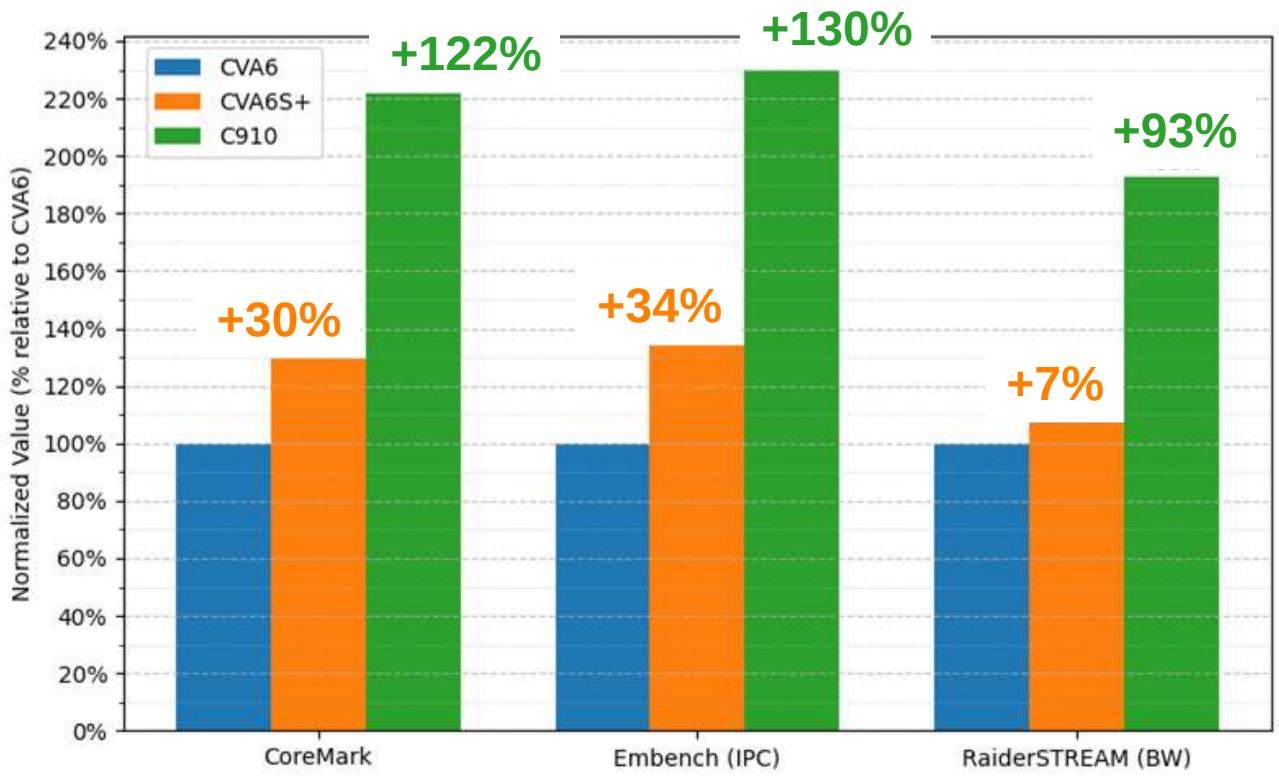
CoreMark®
An EEMBC Benchmark

Embench

RAIDER STREAM

# Performance (IPC) Results

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

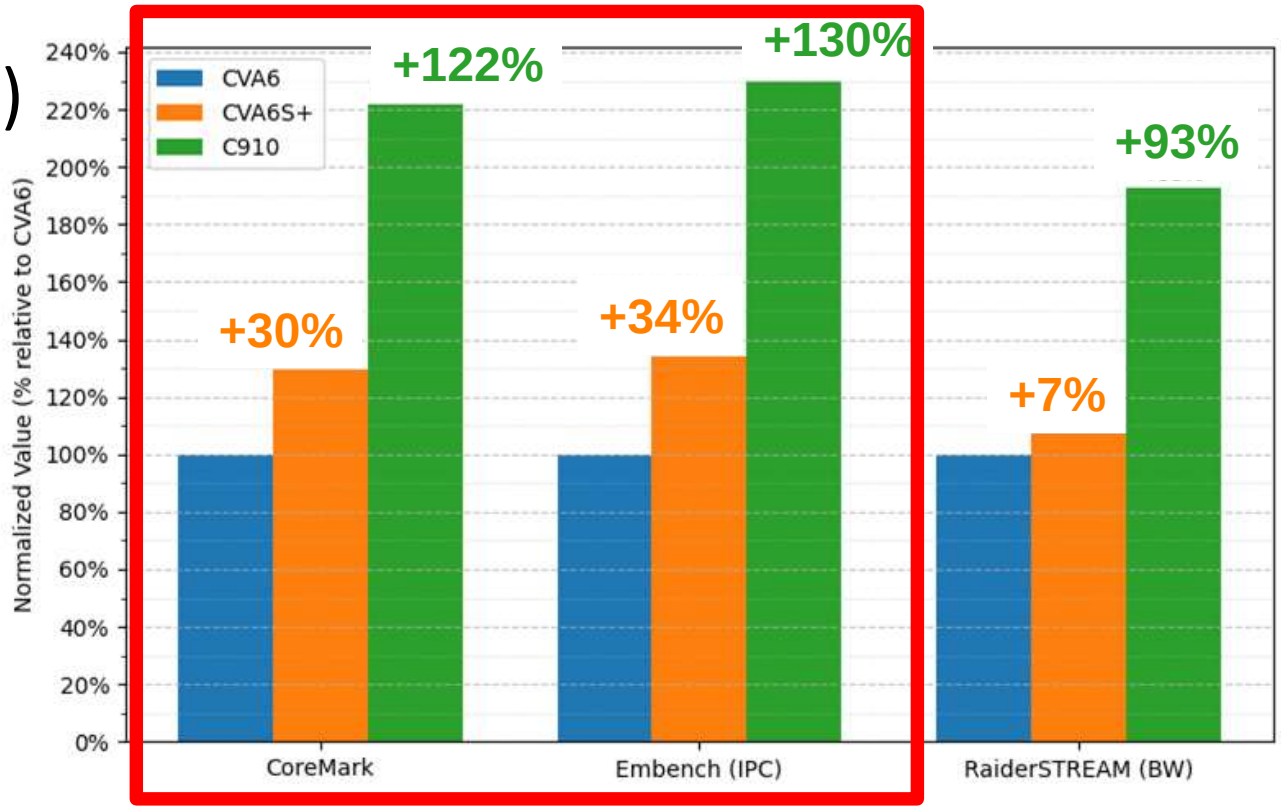



Performance comparison on different benchmarks

# Performance Results

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

- IPC (avg. Coremark & Embench)
  - CVA6S+ ~1.3x CVA6
  - C910 ~2.3x CVA6

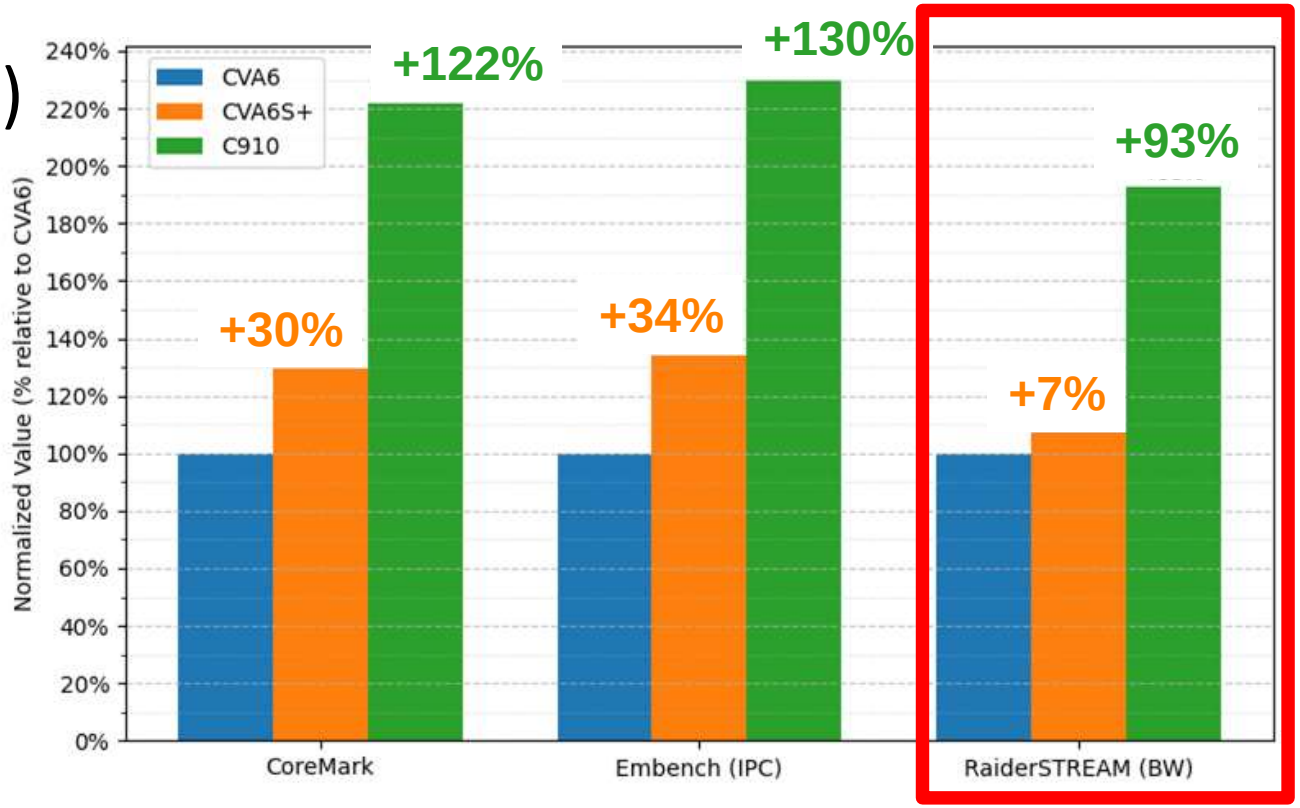



Performance comparison on different benchmarks

# Performance Results

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

- IPC (avg. Coremark & Embench)
  - CVA6S+ ~1.3x CVA6
  - C910 ~2.3x CVA6
- Memory subsystem BW (RaiderSTREAM)
  - CVA6S+ ≈ CVA6 (Same LSU)
  - C910 ~1.9x CVA6




Performance comparison on different benchmarks

# Next, Let's Have a Look at the Backend Implementation

# Backend Flow & Methodology

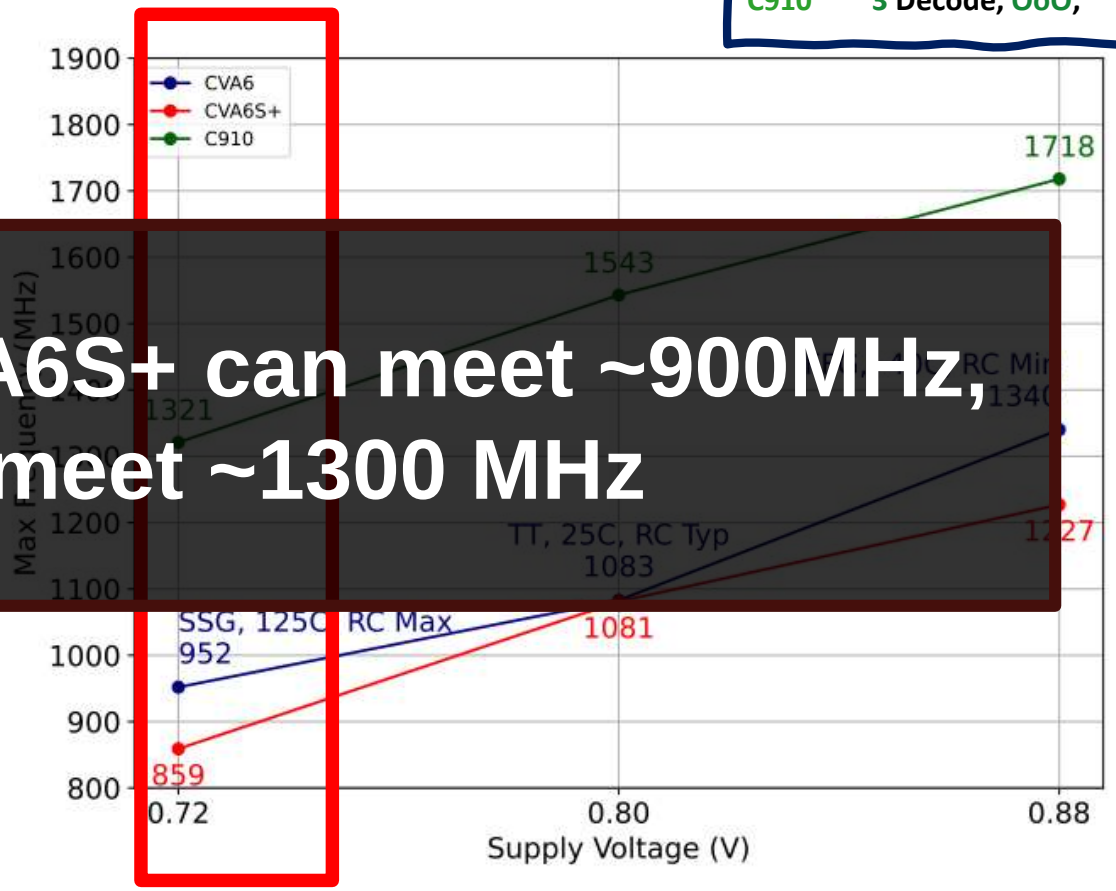
- **Technology**
  - GlobalFoundries 22 FDX (22nm FD-SOI)
  - Timing evaluation at global worst-case corner, 0.72 V, 125 ℃
- **Synthesis**
  - Synopsys Design Compiler 2022.12
- **Physical implementation**
  - Cadence Innovus 20.12
- **Power analysis**
  - Tool: Synopsys PrimeTime 2022.03.
  - At typical corner, 0.80 V, 25℃

# Backend Results: **Timing**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

- Max target frequency (@worst case)
  - CVA6 952 MHz
  - CVA6S+ 859 MHz
  - C910 1321 MHz

**The CVA6 and CVA6S+ can meet ~900MHz, C910 can meet ~1300 MHz**

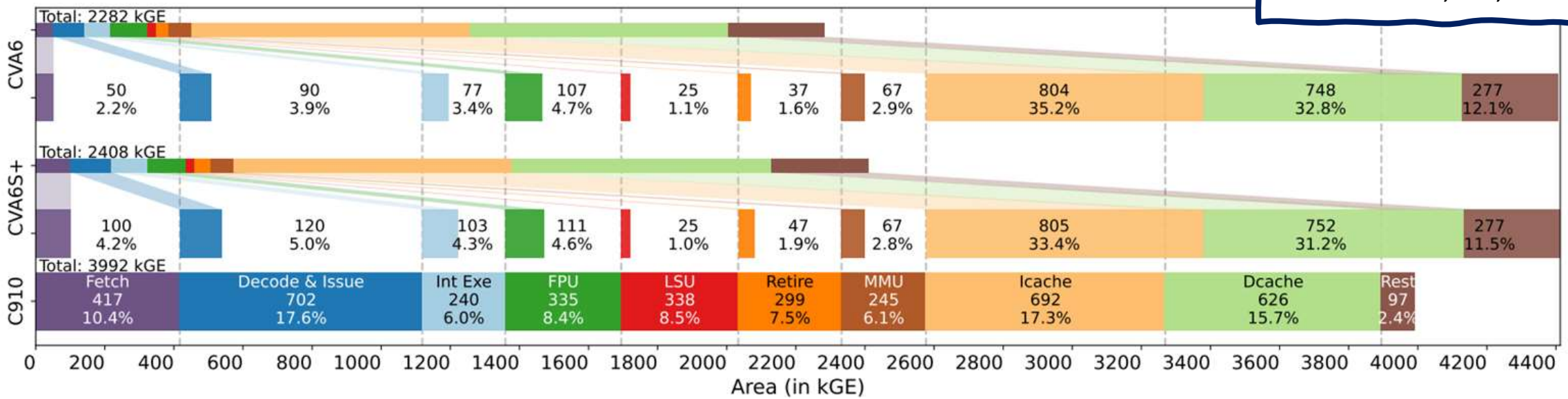



Max Freq under Different Supply Voltages

# Next is the Area Breakdown

# Backend Results: **Area**

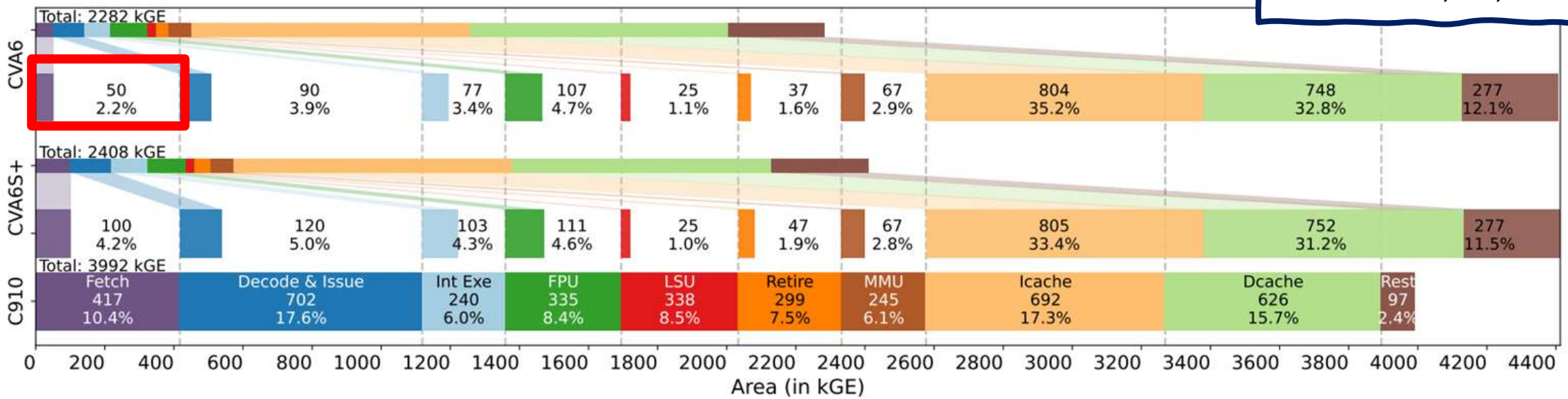
CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage




Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

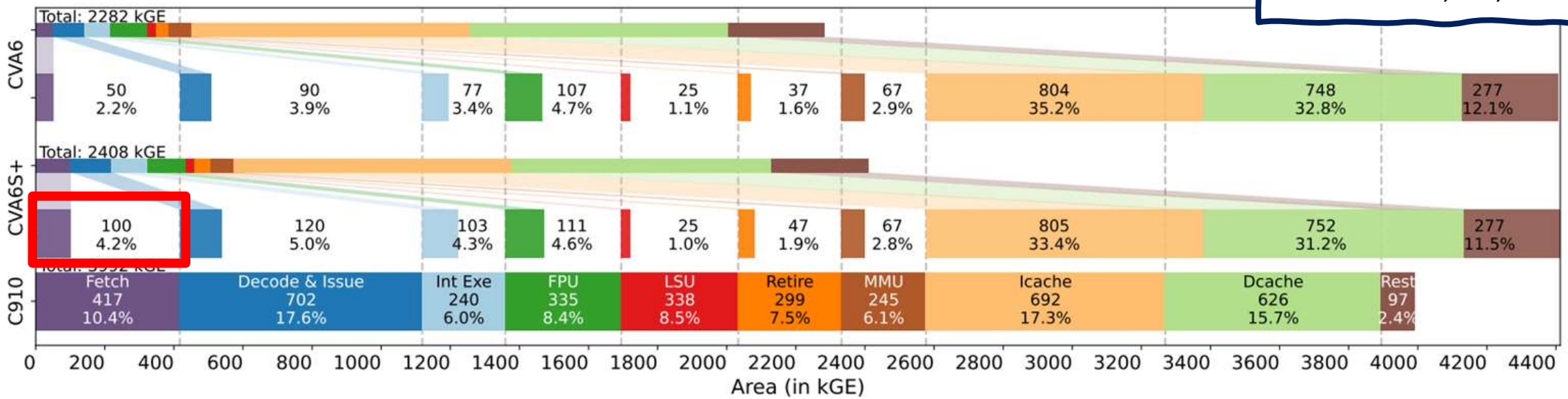



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Fetch Unit**
  - **CVA6** 50 kGE: 128-entry BHT, 32-entry BTB

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

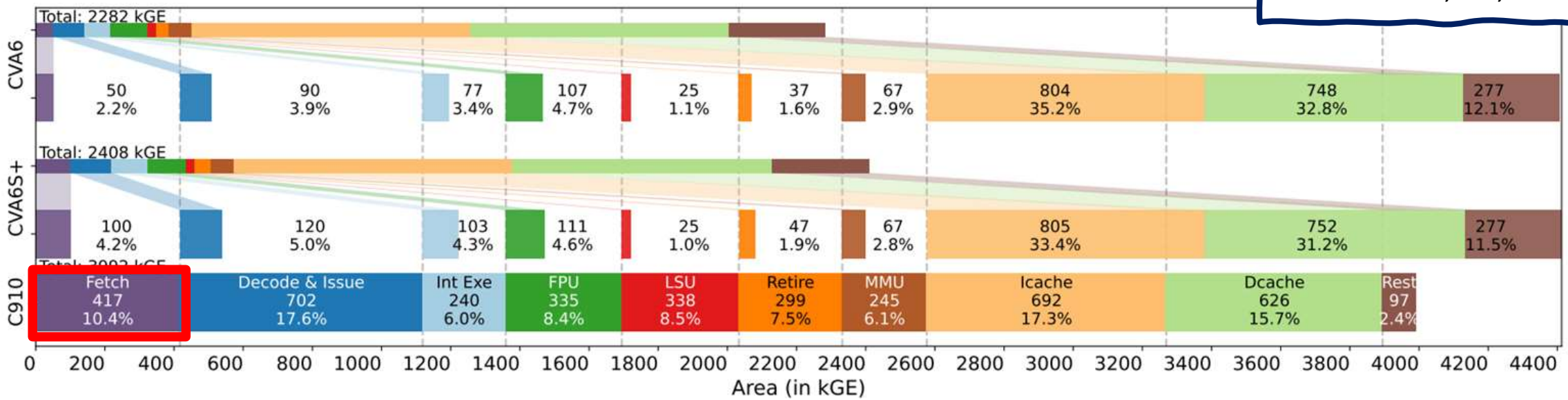



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Fetch Unit**
  - **CVA6** 50 kGE: 128-entry BHT, 32-entry BTB
  - **CVA6S+** 100 kGE (2x): **larger BHT** (w/ private history)

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

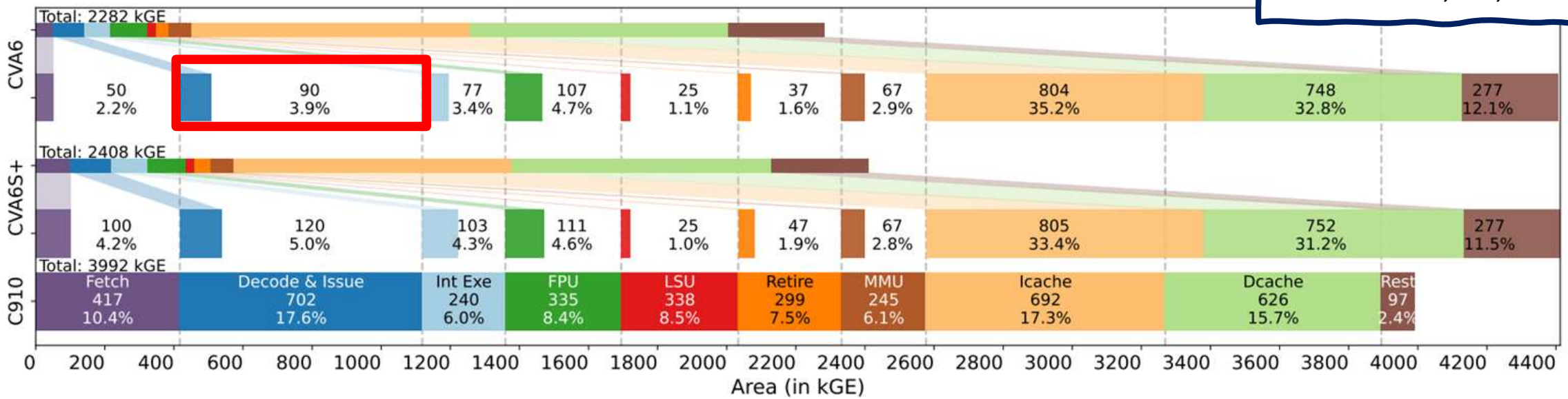



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Fetch Unit**
  - **CVA6** 50 kGE: 128-entry BHT, 32-entry BTB
  - **CVA6S+** 100 kGE (2x): **larger BHT** (w/ private history)
  - **C910** 417 kGE (8.3x): much **larger BHT** (32K-entry) and **BTB** (4K-entry L1 BTB)

# Backend Results: **Area**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

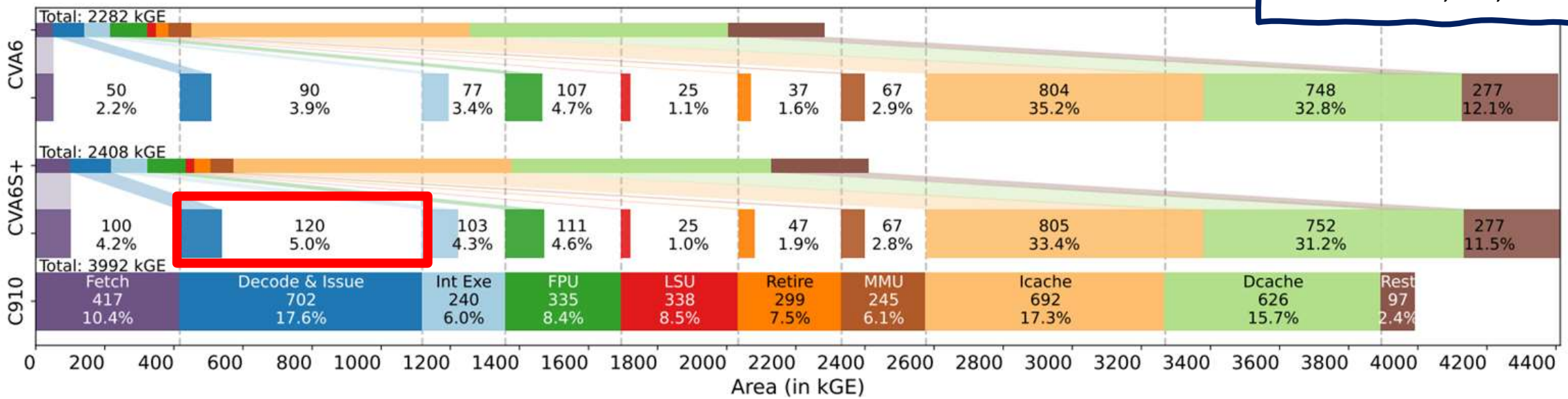



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Decode & Issue Unit**
  - **CVA6** 90 kGE: 1 decoder, in-order issue logic

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

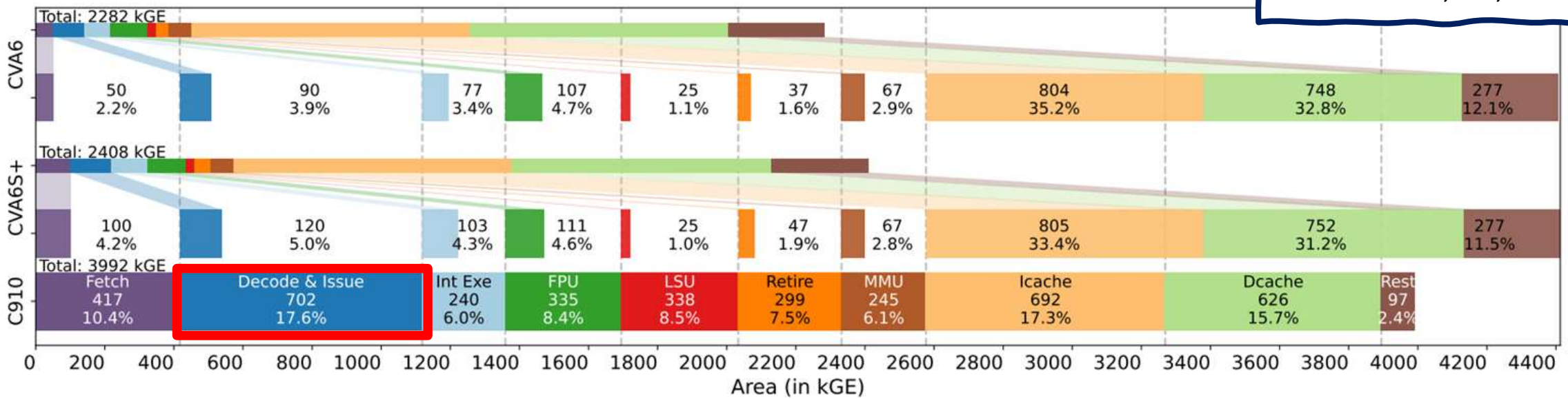



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Decode & Issue Unit**
  - **CVA6** 90 kGE: 1 decoder, in-order issue logic
  - **CVA6S+** 120 kGE (1.3x): **2 decoders**

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

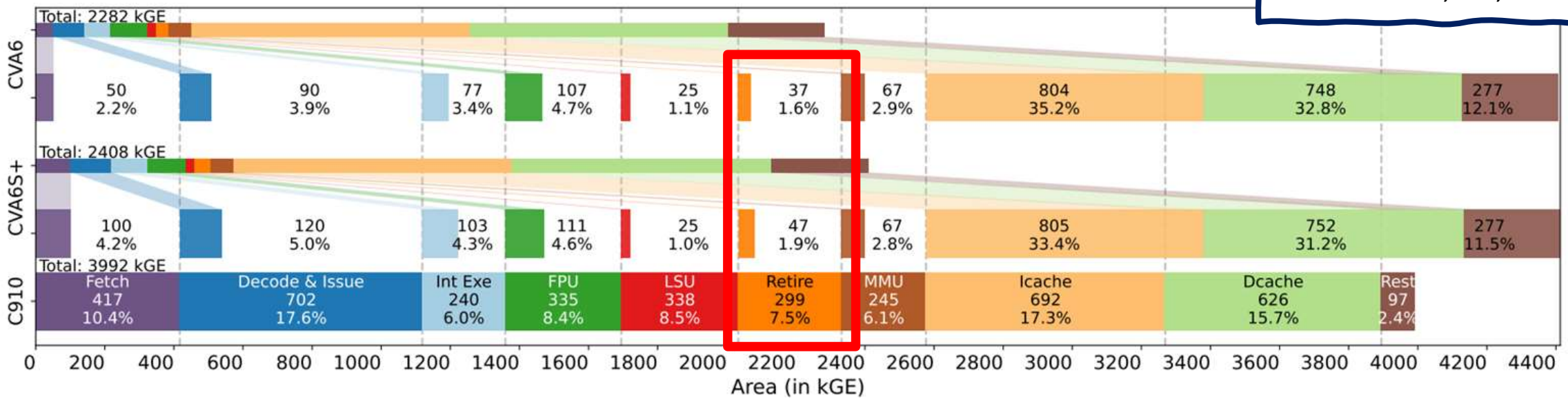



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Decode & Issue Unit**
  - **CVA6** 90 kGE: 1 decoder, in-order issue logic
  - **CVA6S+** 120 kGE (1.3x): **2 decoders**
  - **C910** 702 kGE (7.8x): **3 decoders**, **OoO** issue logic

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

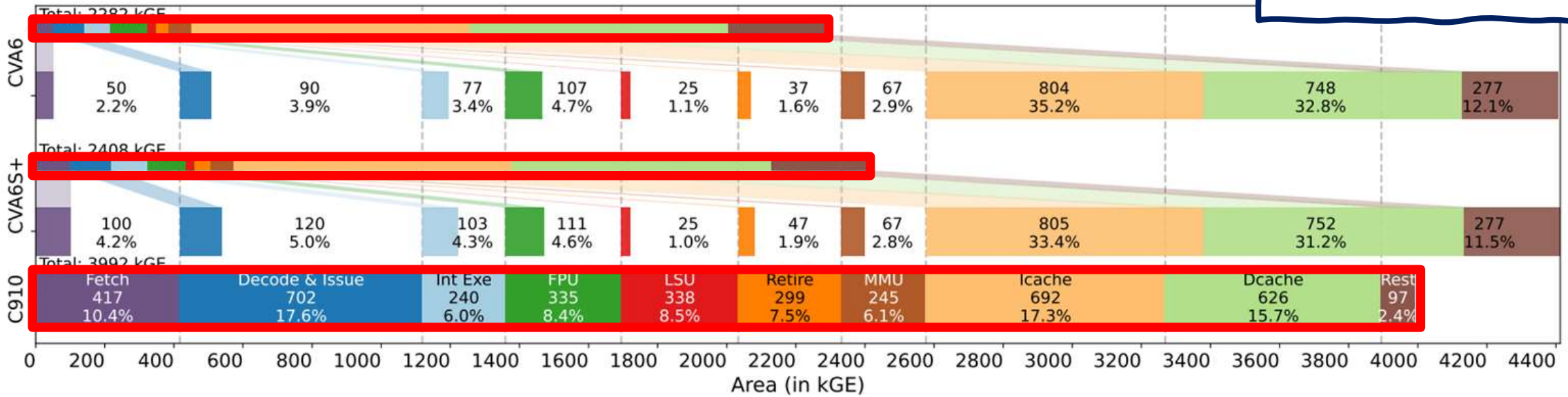



Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Retire Logic**
  - CVA6 37 kGE: 8-entry scoreboard
  - CVA6S+ 47 kGE (1.3x): wider retire logic
  - C910 299 kGE (8.1x): **64-entry ROB** & wider retire logic

# Backend Results: **Area**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage




Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

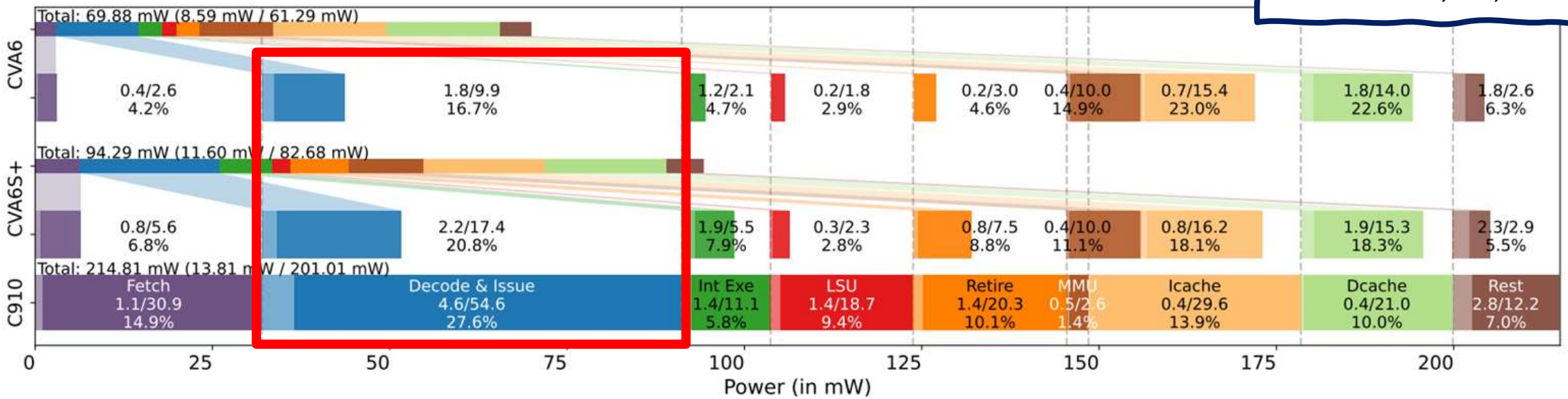
- **In Total**
  - CVA6 2,282 kGE
  - CVA6S+ 2,408 kGE (**1.06x**)
  - C910 3,992 kGE (**1.75x**)

# Then is the Power Breakdown



# Backend Results: **Power**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage

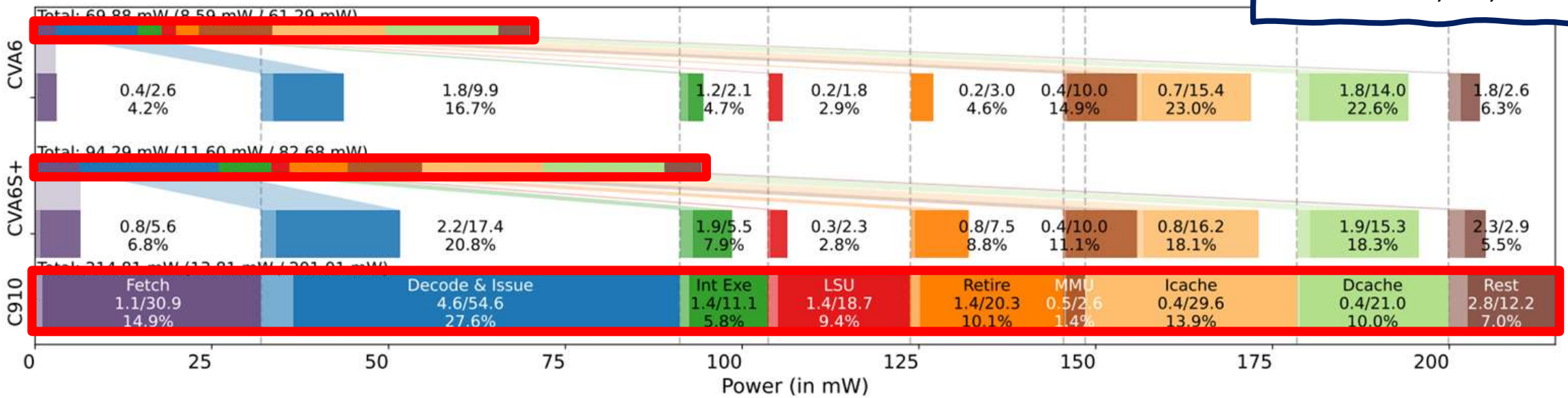



Power Breakdown (Leakage/Dynamic Power) [mW] near the respective maximum target operating frequency:
CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz. Benchmark: matmult-int

- **Issue Logic**
  - **CVA6** 11.7 mW: single-issue, in-order
  - **CVA6S+** 19.6 mW (1.7x): **2-issue**
  - **C910** 59.2 mW (5.1x): **3-issue**, **Out-of-Order**

# Backend Results: **Power**

CVA6 1 Decode, In-Order, 6-stage
CVA6S+ 2 Decode, In-Order, 6-stage
C910 3 Decode, OoO, 12-stage




Power Breakdown (Leakage/Dynamic Power) [mW] near the respective maximum target operating frequency:
CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz. Benchmark: matmult-int

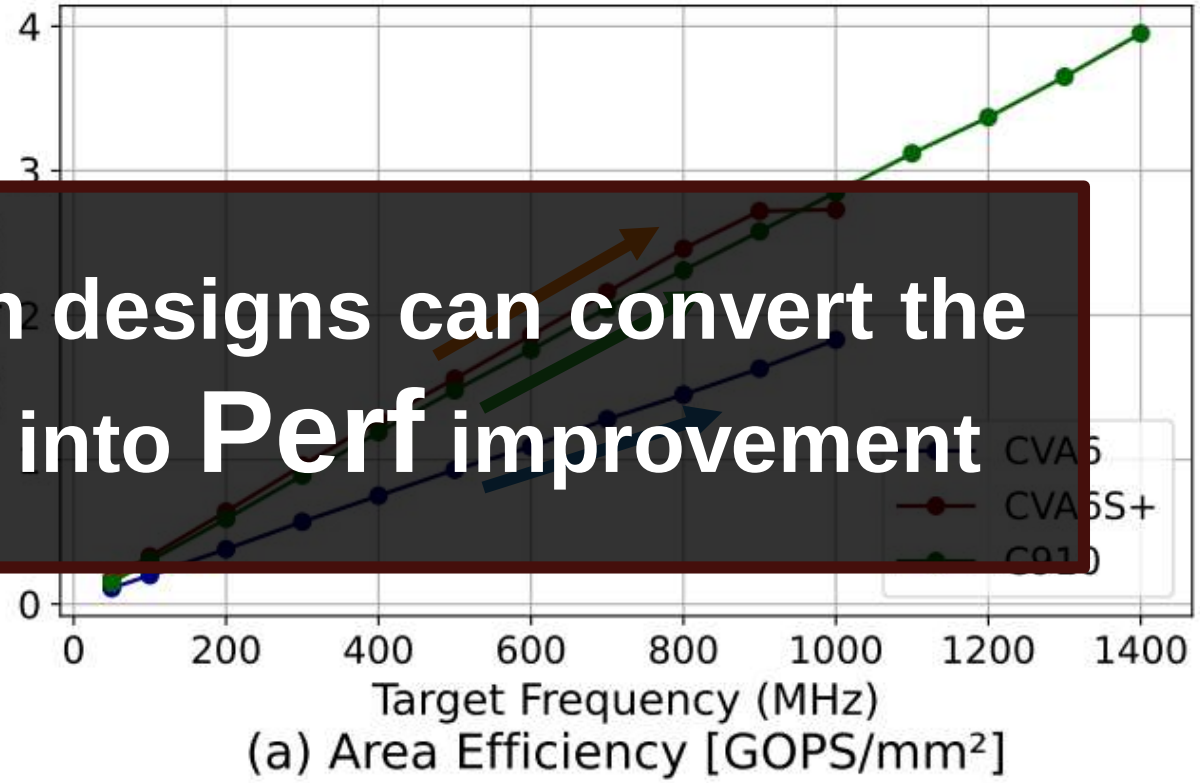
- **In Total**
  - CVA6 69.88 mW
  - CVA6S+ 94.29 mW (**1.35x**)
  - C910 214.81 mW (**3.1x**)

# Putting it all together: **Efficiency**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Area Efficiency (GOPS/mm²)**
  - CVA6S+ leads, notably outperforming CVA6.
  - C910 is [illegible] despite its complexity.
    - The freq gain from deeper pipeline, as well as the IPC gain from Superscalar OoO exe, overweighs increased area

**Reasonable μ-arch designs can convert the increased Area into Perf improvement**

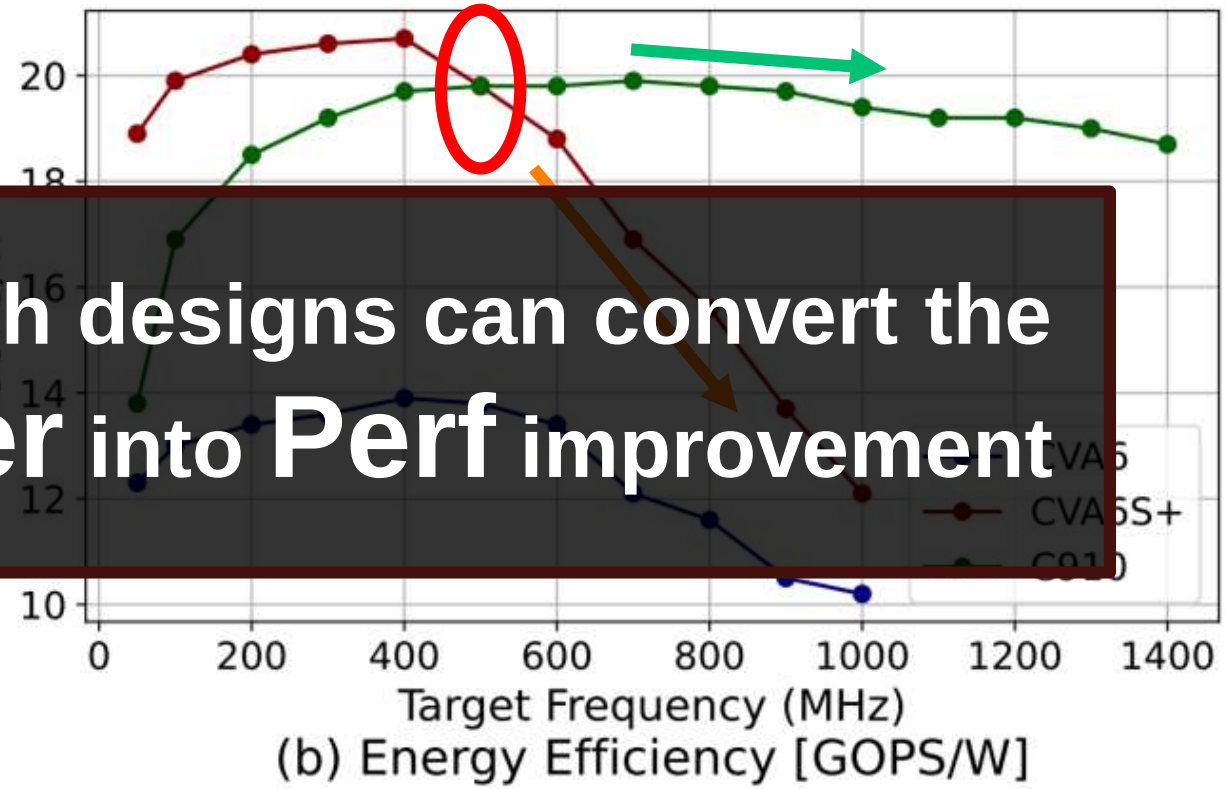



* numbers with the same config of cache

# Putting it all together: **Efficiency**

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- **Energy Efficiency (GOPS/W)**
  - CVA6S+ better when target freq < 500MHz, then drops rapidly,
    - Even though it still has margin from the max feasible frequency (~800MHz)
  - C910 better when target freq > 500MHz
    - Design targets at a higher freq
    - The IPC gain from the complicated structures is comparable to the increased power



(b) Energy Efficiency [GOPS/W]

* numbers with the same config of cache

**Reasonable μ-arch designs can convert the increased Power into Perf improvement**

# Conclusion

| | |
|---|---|
| CVA6 | 1 Decode, In-Order, 6-stage |
| CVA6S+ | 2 Decode, In-Order, 6-stage |
| C910 | 3 Decode, OoO, 12-stage |

- Detailed and **fair** comparative **analysis** of **PPA**, and **Energy Efficiency** across three RISC-V cores designed with **different μ-architectures**.
- Introduced **CVA6S+**: extended **Superscalar** variant of **CVA6**, achieving an avg. **34.4%** [illegible] over CVA6 on the Embench-IoT suite.
- Introduced fully **RISC-V-compliant C910** variant **Superscalar OoO** core, featuring RISC-V standard debug, interrupt, and memory interfaces. Achieves an avg. **130%** better performance over **CVA6** on the Embench-IoT suite.
- **CVA6S+** achieves the **Best Area Efficiency**.
- Beyond Higher frequency **(500 MHz)**, the **OoO C910** core **Surpasses In-Order** cores in **Energy Efficiency**, which challenges the conventional belief that OoO cores are inherently less energy-efficient than in-order cores.

**Despite the higher Area & Power overhead, OoO Superscalar cores can remain competitive in Area & Energy Efficiency**

github.com/pulp-platform/pulp-c910
github.com/pulp-platform/cva6
github.com/pulp-platform/cheshire

**PULP Platform**

Open Source Hardware, the way it should be!


| | |
|---|---|
| **Zexin Fu** | zexifu@iis.ee.ethz.ch |
| **Riccardo Tedeschi** | riccardo.tedeschi6@unibo.it |
| **Gianmarco Ottavi** | gianmarco.ottavi2@unibo.it |
| **Nils Wistoff** | nwistoff@iis.ee.ethz.ch |
| **César Fuguet** | cesar.fuguet@univ-grenoble-alpes.fr |
| **Davide Rossi** | davide.rossi@unibo.it |
| **Luca Benini** | lbenini@iis.ee.ethz.ch |




**Institut für Integrierte Systeme – ETH Zürich**
Gloriastrasse 35
Zürich, Switzerland

**DEI – Universitá di Bologna**
Viale del Risorgimento 2
Bologna, Italy


@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

# Contributions

① We conduct a detailed and **fair** comparative **analysis** of **Performance**, **Area**, **Power**, and **Energy Efficiency**

→ Three cores designed with different micro-architectures:

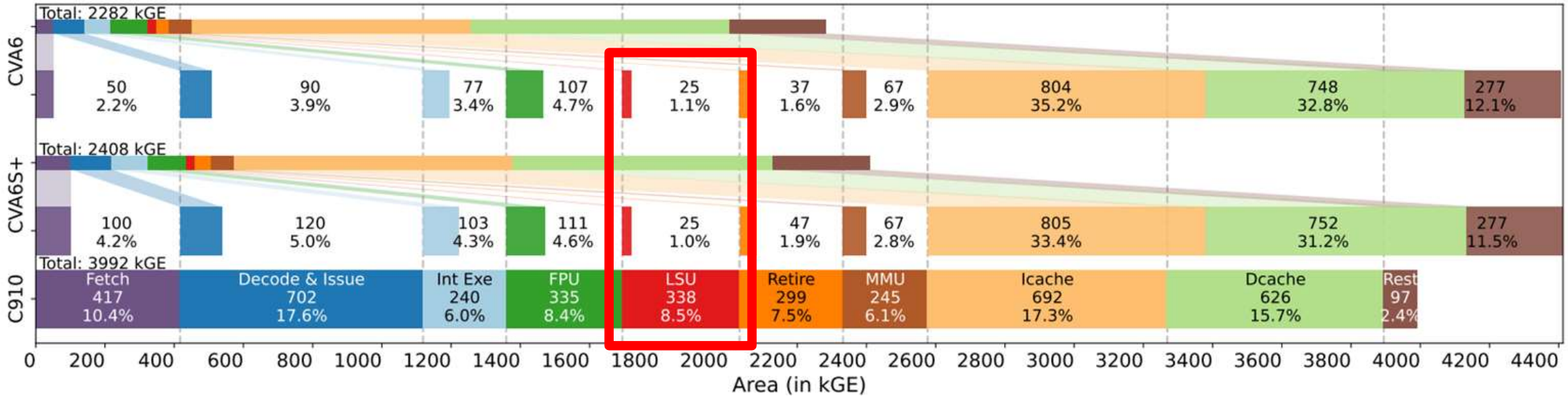
- CVA6, a single-issue, scalar core
- ② We present **CVA6S+**, the extended version of the **Superscalar** variant of CVA6, achieving a **34.4%** performance improvement over CVA6 on the Embench-IoT suite.
- ③ We present an **Open-Source**, fully **RISC-V-compliant** C910-based **Superscalar OoO** core, featuring standard debug, interrupt, and memory interfaces.

→ With the same:

- Base ISA (RV64IMAFDC)
- SoC environment
- Technology node (GF22 FDX)
- EDA tools
- Implementation methodology
- Benchmark binaries

github.com/pulp-platform/pulp-c910

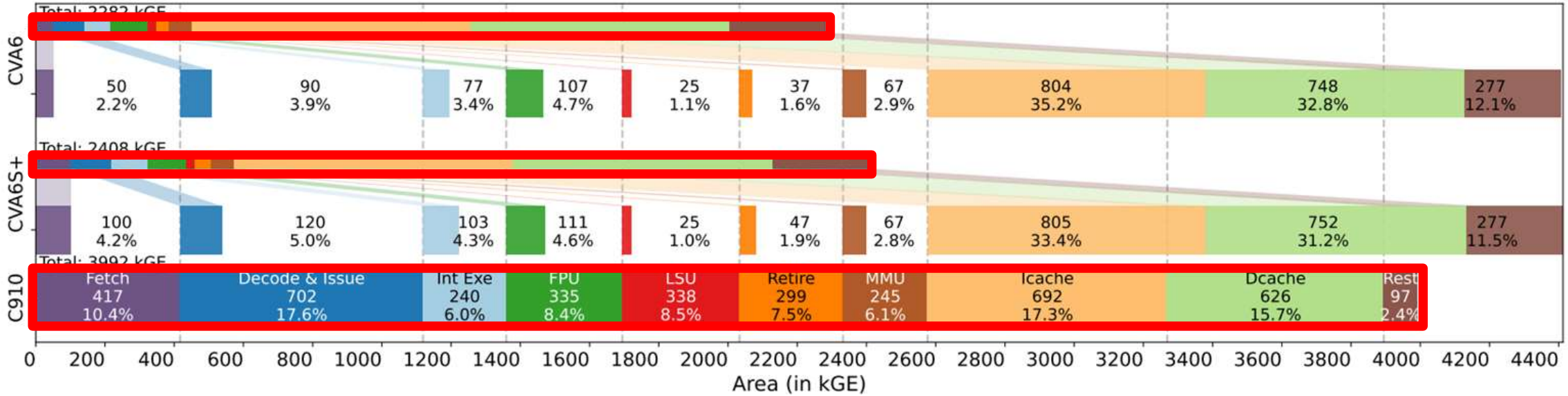
# Backend Results: Area




Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **Load Store Unit**
  - **CVA6** 25 kGE: 1-entry load unit & 8-entry store-queue
  - **CVA6S+** Same as CVA6
  - **C910** 338 kGE (13.5x): **16-entry load queue** & **12-entry store queue**

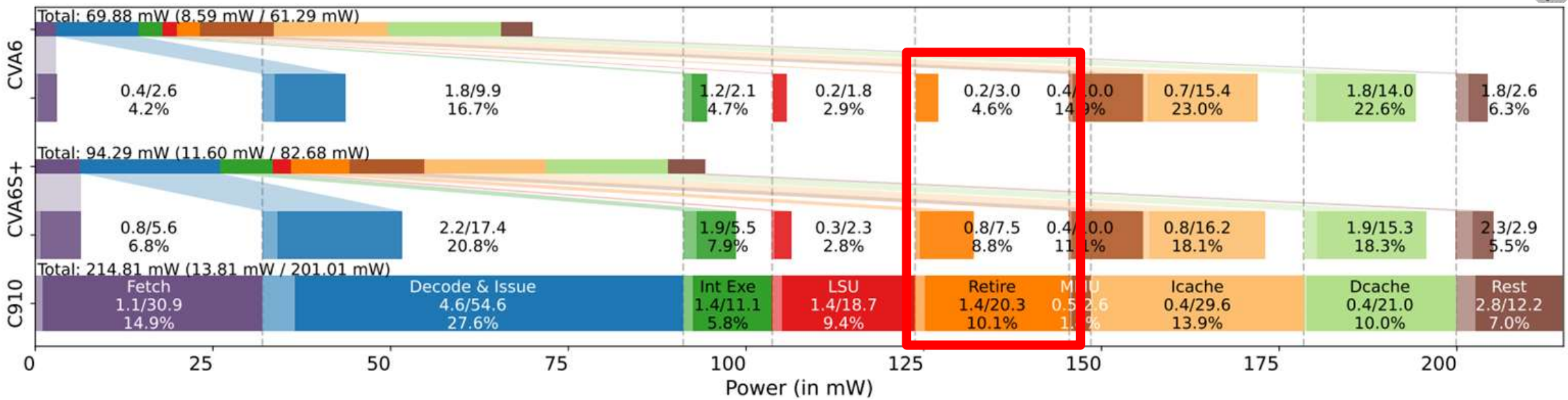
# Backend Results: Area




Area Breakdown [kGE] near the respective maximum target operating frequency: CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz.

- **In Total**
  - **W/** 64 KB, 2-way I-cache & D-cache
    - **CVA6:** 2,282 kGE
    - **CVA6S+:** 2,408 kGE (**1.06x**)
    - **C910:** 3,992 kGE (**1.75x**)
  - **W/o** cache
    - **CVA6:** 730 kGE
    - **CVA6S+:** 851 kGE (**1.17x**)
    - **C910:** 2,674 kGE (**3.7x**)

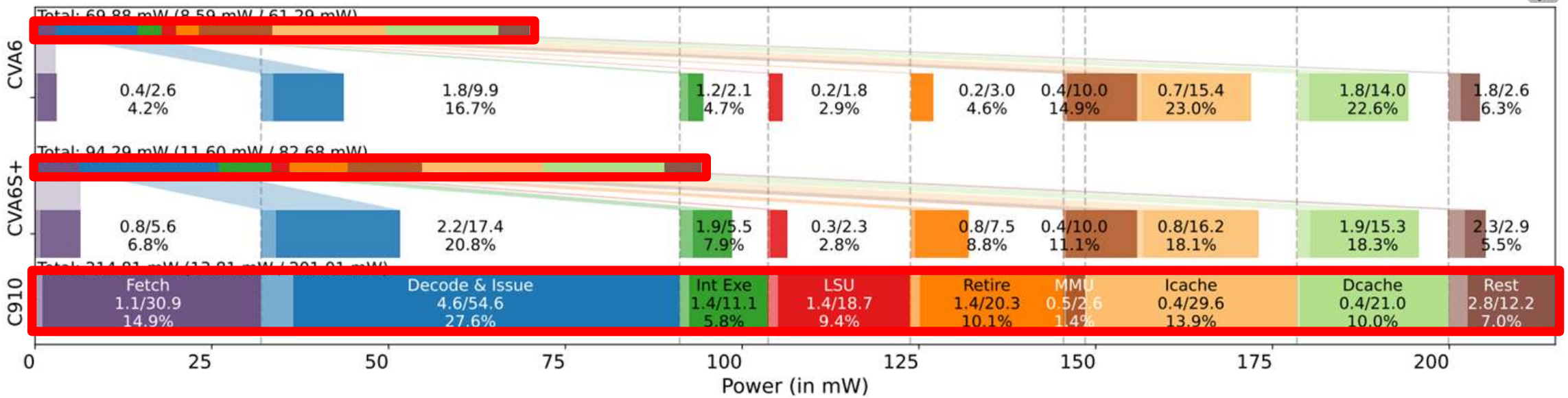
# Backend Results: Power




Power Breakdown (Leakage/Dynamic Power) [mW] near the respective maximum target operating frequency:
CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz. Benchmark: matmult-int

- **Retire Logic**
  - **CVA6** 3.2 mW: 2 retire per cycle, but the second slot is rarely utilized
  - **CVA6S+** 8.3 mW (2.6x): **2-retire**
  - **C910** 21.7 mW (6.8x): **3-retire**

# Backend Results: Power

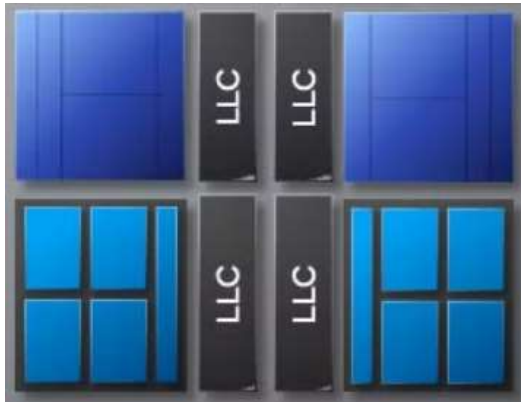



Power Breakdown (Leakage/Dynamic Power) [mW] near the respective maximum target operating frequency:
CVA6 and CVA6S+ at 900MHz, while C910 at 1300 MHz. Benchmark: matmult-int

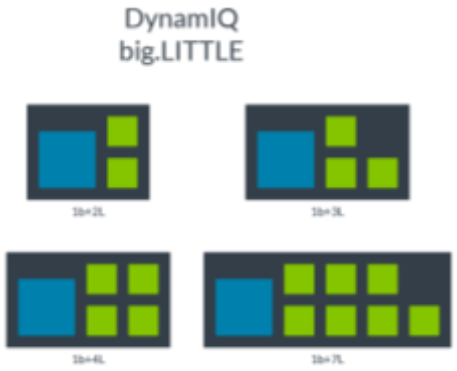
- **In Total**
  - **W/** 64 KB, 2-way I-cache & D-cache
    - **CVA6:** 69.88 mW
    - **CVA6S+:** 94.29 mW (**1.35x**)
    - **C910:** 214.81 mW (**3.1x**)
  - **W/o** cache
    - **CVA6:** 37.98 mW
    - **CVA6S+:** 60.09 mW (**1.58x**)
    - **C910:** 163.41 mW (**4.3x**)

# Energy Efficiency: A Rising Concern

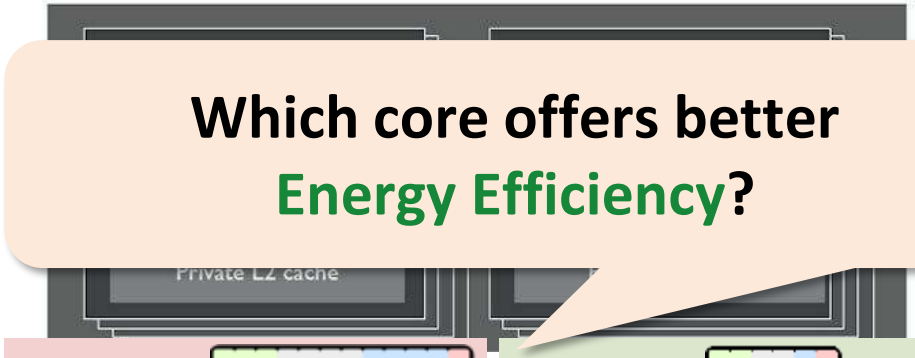
- **Prevailing Energy Efficient Design: Heterogeneous CPU cluster**
  - High Performance Core + Low Power Core
  - Balance **Peak Performance** with Power-Efficient Handling of **Lightweight Tasks**




Intel Performance Cores and Efficient Cores




ARM big.LITTLE design

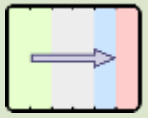



**Which core offers better Energy Efficiency?**

**Core A:**

**Wide Decode Width**
**Out-of-Order**
**>10 Pipeline Stages**

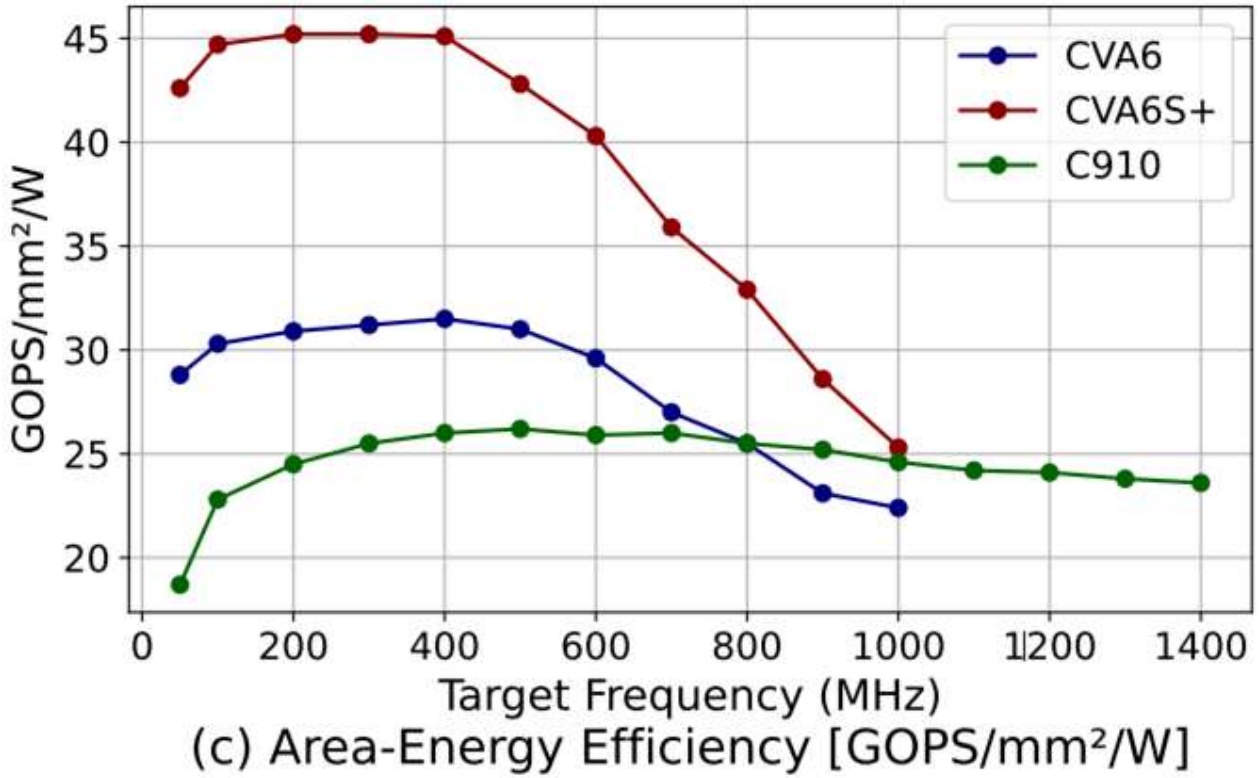
**Core B:**



**Narrow Decode Width**
**In-Order**
**3-10 Pipeline Stages**

# Putting it all together: Efficiency

- **Area-Energy Efficiency (GOPS/mm2/W)**
  - CVA6S+ always the best




* numbers with the same config of cache